

VSB – TECHNICAL UNIVERSITY OF OSTRAVA
FACULTY OF ECONOMICS

THESIS OF DOCTORAL DISSERTATION

Technical and Fundamental Features analysis for Stock Market Prediction with Data Mining Methods

Author: Sasan Barak

Advisor: Prof. Tomáš Tichý

Co-advisor: Prof. Sergio Ortobelli Lozza

Study program: Economic Policy and Administration

Field of study: Finance

VSB – TECHNICAL UNIVERSITY OF OSTRAVA
– FACULTY OF ECONOMICS -
and
UNIVERSITÀ DEGLI STUDI DI BERGAMO

Doctoral Thesis
Technical and Fundamental Features analysis for Stock Market
Prediction with Data Mining Methods

Author:
Sasan Barak

Supervisor:
Prof. Tomas Tichý
Co-supervisor:
Prof. Sergio Ortobelli Lozza

*A thesis submitted in fulfilment of the requirements
for the degree of Doctor of Philosophy in*

Ph.D. School in Economic Policy and Administration - Finance

and

Ph.D. School in Analytics for Economics and Business

May 2019

Abstract

Predicting stock prices is an essential objective in the financial world. Forecasting stock returns and their risk represents one of the most critical concerns of market decision makers. This thesis investigates the stock price forecasting with three approaches from the data mining concept and shows how different elements in the stock price can help to enhance the accuracy of our prediction. For this reason, the first and second approaches capture many fundamental indicators from the stocks and implement them as explanatory variables to do stock price classification and forecasting. In the third approach, technical features from the candlestick representation of the share prices are extracted and used to enhance the accuracy of the forecasting. In each approach, different tools and techniques from data mining and machine learning are employed to justify why the forecasting is working.

Furthermore, since the idea is to evaluate the potential of features in the stock trend forecasting, therefore we diversify our experiments using both technical and fundamental features. Therefore, in the first approach, a three-stage methodology is developed while in the first step, a comprehensive investigation of all possible features which can be effective on stocks risk and return are identified. Then, in the next stage, risk and return are predicted by applying data mining techniques for the given features. Finally, we develop a hybrid algorithm, based on some filters and function-based clustering; and re-predicted the risk and return of stocks.

In the second approach, instead of using single classifiers, a fusion model is proposed based on the use of multiple diverse base classifiers that operate on a common input and a meta-classifier that learns from base classifiers' outputs to obtain a more precise stock return and risk predictions. A set of diversity methods, including Bagging, Boosting, and AdaBoost, is applied to create diversity in classifier combinations. Moreover, the number and procedure for selecting base classifiers for fusion schemes are determined using a methodology based on dataset clustering and candidate classifiers' accuracy.

Finally, in the third approach, a novel forecasting model for stock markets based on the wrapper ANFIS (Adaptive Neural Fuzzy Inference System) – ICA (Imperialist Competitive Algorithm) and technical analysis of Japanese Candlestick is presented. Two approaches of Raw-based and Signal-based are devised to extract the model's input variables and buy and sell signals are considered as output variables.

To illustrate the methodologies, for the first and second approaches, Tehran Stock Exchange (TSE) data for the period from 2002 to 2012 are applied, while for the third approach, we used General Motors and Dow Jones indexes.

Keyword: Stock Market; Data Mining; Machine Learning; Feature Selection; Technical and Fundamental Features.

Acknowledgements

First and foremost, I would like to thank my supervisors Prof. Tomas Tichý and Prof. Sergio Ortobelli Lozza for giving me the opportunity to carry out this project and for all their invaluable advices, supports, and encouragements.

I have been exceedingly lucky in the support, knowledge, and training skills which I have received from those around me during this thesis, and in addition to my supervisors, there are some people who deserve specific mention. I would like to express my gratitude to the former directors of the Ph.D. students at the University of Bergamo, Prof. Marida Bertocchi (R.I.P) for her constant motivation and encouragement and Prof. Dr. Ing. Zdenek Zmeskal for the support and suggestion received. For this dissertation, I would like to thank my reading committee members and all my Professors.

I would like to thank all my family for all their love and continuous encouragement specially my parents who raised me with a love of science and supported me in all my pursuits. I am very grateful to my loving, supportive, encouraging, and patient wife Masoume whose faithful support during the stages of this Ph.D. is so appreciated.

Finally, I would like to thank my friends and colleagues at the University of Bergamo, and the Technical University of Ostrava for their friendship and intellectual camaraderie over the past few years.

Thank you.

Contents

1. Introduction.....	8
2. Generic CRISP-DM Reference Model	13
2.1. Business Understanding.....	13
2.2. Data Understanding	14
2.3. Data Preparation	14
2.3.1. Feature Selection	14
2.4. Modeling.....	16
2.4.1. Classification and forecasting methods	16
2.4.2. Clustering.....	19
2.5. Evaluation	19
2.6. Deployment.....	20
3. Developing an approach to evaluate stocks by forecasting effective features with data mining methods	21
3.1. Introduction.....	21
3.1.1. First stage: Developing financial database	24
3.1.2. Second stage: Risk and return prediction with classification methods	27
3.1.3. Third stage: Hybrid feature selection.....	29
3.2. Experimental results and analysis.....	31
3.2.1. Data pre-processing	32
3.2.2. Comparison of algorithms	33
3.2.3. Prediction after hybrid Feature selection	37
3.3. Discussion.....	40
3.4. Conclusions.....	42
4. Fusion of Multiple Diverse Predictors in Stock Market	44
4.1. Introduction.....	44
4.2. Literature Review	46
4.2.1. Stock prediction with classifier ensembles	46
4.2.2. Fusion in ensembles scheme.....	47
4.2.3. Diversity and accuracy creation.....	48
4.3. Proposed Multi-Classifer Ensemble System (MCS)	48
4.3.1. Phase one: Base classifier generation	49
4.3.2. Phase two: Classifier selection	52
4.3.3. Phase three: Fusion	57
4.4. Experimental Results.....	58
4.4.1. Individual classifiers	59
4.4.2. Selection of classifier sets.....	60

4.4.3. Stock return and risk predictions with the proposed fusion scheme	61
4.5. Discussion.....	63
4.6. Conclusions and Future Research.....	68
5. Wrapper ANFIS-ICA method to do stock market timing and feature selection based on Japanese Candlestick	70
5.1. Introduction.....	70
5.2. The background	72
5.2.1. Wrapper feature selection method	72
5.2.2. Adaptive Neural Fuzzy Inference System (ANFIS)	73
5.2.3. Imperialist Competitive Algorithm (ICA)	75
5.3. The proposed model	77
5.3.1. Adaptive TAs and ANFIS.....	77
5.3.2. Model.....	79
5.4. Empirical results	84
5.4.1. Input data	84
5.4.2. Results.....	85
5.5. Conclusion	92
6. Conclusion and Future Research	94
Literature	97
Appendix A.....	104
Appendix B.....	106
Appendix C.....	107
List of Tables	108
Summary list of my publications during the PhD.....	109
Declaration of Authorship	110

Chapter 1

Introduction

One of the most critical concerns of market practitioners is future information of the companies which offer stocks. A reliable prediction of the company's financial status provides a situation for the investor to more confident investments and gaining more profits (C. F. Huang, 2012). Accurately prediction of stocks' prices has positive effects on the organizations' financial stability (Asadi, Hadavandi, Mehmanpazir, & Nakhostin, 2012). Since the financial market is complex and has non-linear dynamic systems, its prediction is challenging (C.-L. Huang & Tsai, 2009).

In the different aspect, the steady and fantastic progress of computer hardware technology in the past decades has led to abundant supplies of powerful and affordable computers, data collection equipment, and storage media. Besides, enhancements of artificial intelligence (AI) provide opportunities to create meaningful information from vast data efficiently and to analyze them precisely. In the financial sector which is one of the most important beneficiaries of AI advantages, researchers and practitioners assist in utilizing machine learning (ML) and data mining tools, as a subset of AI, to identifying the behavior of markets and price movements to optimize investment portfolios.

Data mining as a new growing interdisciplinary branch of science that combines various fields like statistics, machine learning, and parallel processing to extract knowledge from data flooding around organizations, governments, and individuals is a useful approach to predict future information of the companies. Potent data analysis and machine learning models along with technology, provide computerized trading systems can help to reduce the complexity of decision making in the financial portfolio management. In the asset management criteria, ML can enable individual and institutional investors to assess the level of risk and upcoming return for a specific portfolio of financial instruments. Additionally, with using historical data of market performance, AI can help investors to optimum their portfolio under different economic conditions. Moreover, recommending the best opportunity for investing by integrating the massive amount of data which comes from multiple sources including financial reporting institutions, news agencies, and economic firms, could be considered as benefits of the AI in finance. Furthermore, the analysis process of data and its interpretation could happen in an automated manner like robo-advisers or algorithmic robots.

Algorithm trading with the automation of trading based on the combination of mathematical models and human insights has shown rapid growth in the finance sector and shown an acceptable performance in comparison with traditional financial trading as it is shown in Figure 1.1. The superiority of the Eureka hedge

AI hedge fund as a representative of hedge funds that utilize AI in their trading process is clear to the HFRI index which is a global, equal-weighted index of the largest hedge funds that report to the HFR¹ Database.

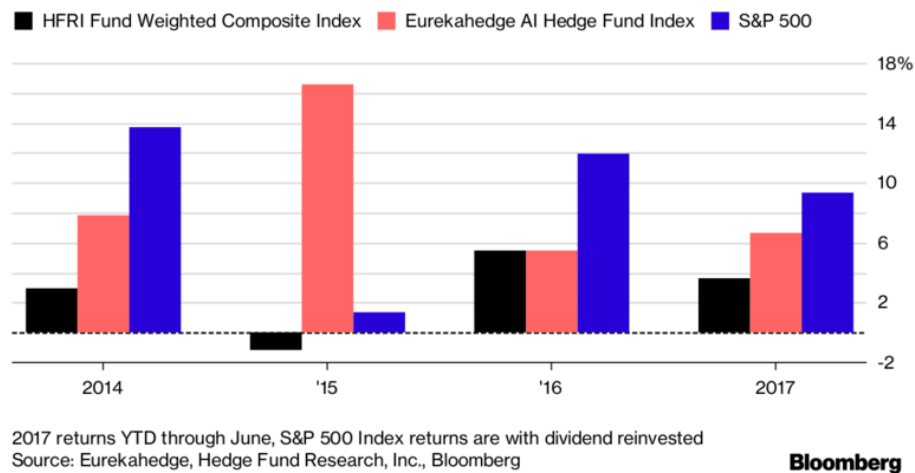
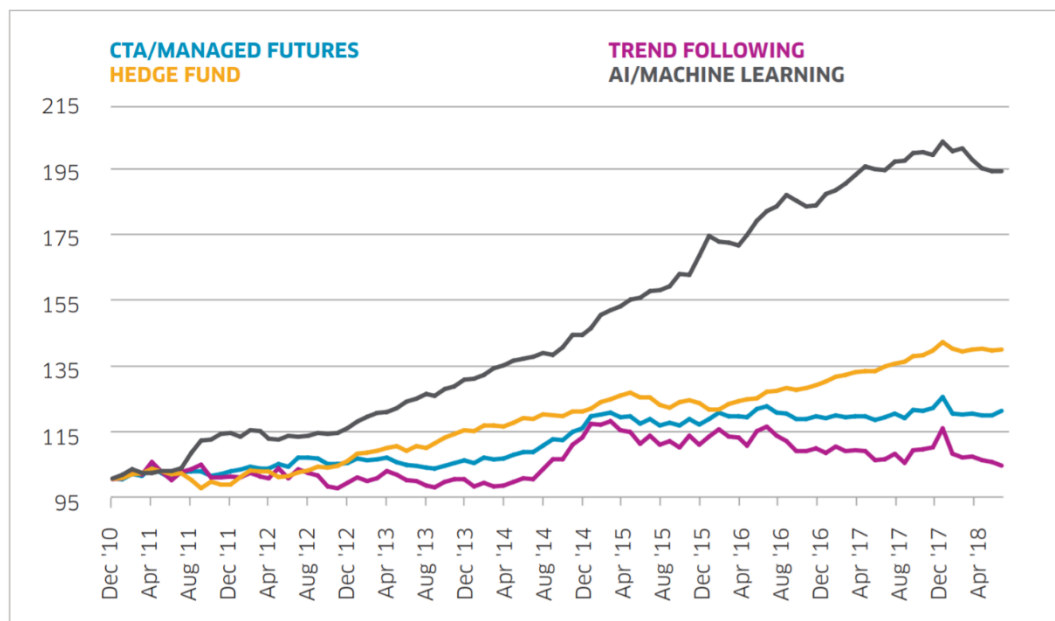


Figure 1.1 Machine learning's gains

Similarly, as it is shown in Figure 1.2, among the different core concept of trading strategies which are implemented by hedge funds, the AI/machine learning based strategies showed a significant privilege in comparison with other hedge funds during December 2010 to 2016.



Source: EurekaHedge. Data Range: Dec '10 - Jul '18

Figure 1.2 EurekaHedge indexes' performance

Besides, prediction of credit scores, risk assessment, and even bankruptcy potentials are examples of significant applicability of data mining in the field of finance. One can refer to different studies about share gaining and return prediction, for example, time series stock price prediction model (Araújo & Ferreira,

¹ Hedge Fund Research, Inc.

2013), buy-hold-sell prediction model (Wu, Yu, & Chang, 2014; X. Zhang, Hu, Xie, Zhang, et al., 2014), index prediction model with Adaptive neuro fuzzy inference system (ANFIS) (Svalina, Galzina, Lujić, & Šimunović, 2013) or Index prediction with support vector regression (SVR) (Kao, Chiu, Lu, & Chang, 2013), and profit gaining (Ng, Liang, Li, Yeung, & Chan, 2014).

One of the core concepts in the machine learning modeling process is the feature selection that impressively affects the performance of the model. Better feature selection reduces over-fitting, improves accuracy, and optimizes time consumption for model preparation by decreasing model complexity (Cai, Luo, Wang, & Yang, 2018). In the stock market price forecasting, there are various features which come from the two most important types of sources, including fundamental analysis and technical analysis. In this thesis, the potential tools of data mining area are used for the forecasting stocks and shares' prices and future trends using a diverse pool of fundamental and technical features.

However, there are different approaches to financial forecasting in general and stock market price forecasting in particular, such as implementing fundamental analysis, technical analysis, and news via econometric or machine learning algorithms (Atsalakis, Dimitrakakis, & Zopounidis, 2011; Beyaz, Tekiner, Zeng, & Keane, 2018; Kar, Das, & Ghosh, 2014), while in this thesis we will go through all of these methodologies. A brief presentation about the technical and fundamental analyses are presented below:

Technical analysis (TA): evaluates securities using statistics such as past price and volume that are generated by market activities (Barak, Dahooie, & Tichý, 2015). The major criticism of TA is that it only considers the price movement and ignores the fundamental factors related to the company. Moreover, TA takes a comparatively short-term approach to analyze the market.

Fundamental analysis: can be used to evaluate a firm's performance and financial status over a period of time by carefully analyzing the institute's financial statement (C.-F. Huang, 2012). Fundamental analysis seeks to find the essential features of stock and market movements. The logic behind the fundamental analysis is that if a company has a proper fundamental strength, then long-term stock investment in the company will be more secure and stable. Thus, the stocks of these fundamentally strong companies, which are making money, gaining profit, and growing their businesses, represent an opportunity for a successful investment.

Furthermore, many studies have claimed and verified that feature selection (FS) is the key process in stock prediction modeling (Tsai & Hsiao, 2010). X. Zhang, Hu, Xie, Wang, et al. (2014) use a causal feature selection (CFS) algorithm to find effective features in Shanghai Stock Exchanges. The idea in their model is about Causalities based feature selection algorithm. They assert that CFS represents direct influences between various stock features, while correlation-based algorithms cannot distinguish direct influences from indirect ones. Wu, et al. (2014) use textual and technical features to improve the prediction accuracy of the

stock market. They use the SVR algorithm and trend segmentation method to forecast trends and generate trading signals, respectively. Their feature selection algorithm is stepwise regression analysis. Although there are a variety of studies in the area of feature selection, almost all of them use a single feature selection model. However, in this thesis, two hybrid feature selection methodologies are presented to enhance the efficiency of the prediction.

Although the statistical approaches such as logistic regression and regression analysis are widely applied to forecast the return and risk of stocks, the results of machine learning approaches are generally superior in comparison to statistical methods (Barak & Modarres, 2015; S.-H. Cheng, 2014, 2015). The multiple classifier ensemble systems (MCS), one type of machine learning technique, has recently become the focus of a new methodology for obtaining higher accuracy in predictions. The rationale is that the optimization of a combination of relatively more straightforward predictors appears more convenient than optimizing the design of a single complex predictor (Barak & Sadegh, 2016; Haghighi, Vahedian, & Yazdi, 2011). This combination scheme aims to gain increased precision with proper single classifiers and eliminate the uncorrelated individual classifier errors, which are the errors made by individual classifiers on various parts of input space (Tsai, Hsu, & Yen, 2014). Therefore, the fusion model framework that relies on the combination of multiple dissimilar and diverse classifiers operating on a common input is investigated in this thesis.

This thesis presents a systematic and efficient methodology for comprehensive searching the potential representative features on the stock market in both fundamental and technical ways and not arbitrarily choosing likely useful features. Investigating each algorithm with a feature-oriented viewpoint indicates the factors which cause the strength and weakness of that algorithm. Therefore, by searching about the property of the database, we can choose a proper algorithm without the implementation of all methods. This idea can be further extended not only in quantitative investment, but also in other fields of studies where expert systems and machine learning techniques are used.

The following of the thesis consists of six chapters about using technical and fundamental features for stock market prediction with different algorithms in data mining. Fundamental features are the primary focus of investors to bring their funds for creating valuable portfolios. Although technical features are implemented to identify opportunities by focusing on price movements instead of concentrating on stocks' intrinsic value. Accordingly, this thesis exploits different types of financial data set and established three aspects of stock market forecasting via a different combination of a diverse feature engineering in the finance dataset and machine learning models.

In the second chapter, a background and preliminary tools in the data mining area are described based on the generic CRISP-DM reference model.

Chapter three is devoted to using fundamental features to predict the stocks' return and risk and proposed a novel hybrid feature selection approach to enhance the efficiency of prediction results. To do this, first through a comprehensive study, the features which can be potentially effective on risk and return were investigated. Then, after developing an appropriate database, the preprocessing of the database step was taken. To predict the real return and risk, 20 and 15 different prediction algorithms were applied respectively. Then, the strength and weakness of each one were investigated by analyzing the size and leaves of tree algorithms or/and "If-Then Rules" gains of rule-based algorithms. In the next step, by using a hybrid feature selection algorithm based on 9 different filter algorithms and function-based clustering method, important features were selected and re-prediction with selected features was performed.

Chapter four is proposed a developed hybrid machine learning approach to solve the same problem in chapter three for the risk and return prediction. Bagging, Boosting, and AdaBoost were applied as three diversity algorithms for generating a pool of classifiers. An empirical study was later undertaken on the Tehran Stock Exchange that compared the performance of diversity algorithms with different sets of classifiers in a fusion system.

Chapter five focuses on the forecasting with the features captured from technical analysis concept with a hybrid algorithm of ICA (an optimization algorithm) and ANFIS (a forecasting model) for stock trend prediction. Technical knowledge usually is elicited by Technical Analysis, and in this study, a Japanese Candlestick chart analysis is used in order to elicit technical knowledge.

Finally, in the last chapter, all the chapters are combined, and a general conclusion is presented.

Chapter 2

Background: The Generic CRISP-DM Reference Model

The CRISP-DM reference model for data mining provides an overview of the life cycle of a data mining project (Shearer, 2000). It contains the phases of a project, their respective tasks, and their outputs. The life cycle of a data-mining project is broken down in six phases which are shown in Figure 2.1. The sequence of the phases is not strict. The arrows indicate only the most important and frequent dependencies between phases, but in a particular project, it depends on the outcome of each phase which phase, or which particular task of a phase, has to be performed next.

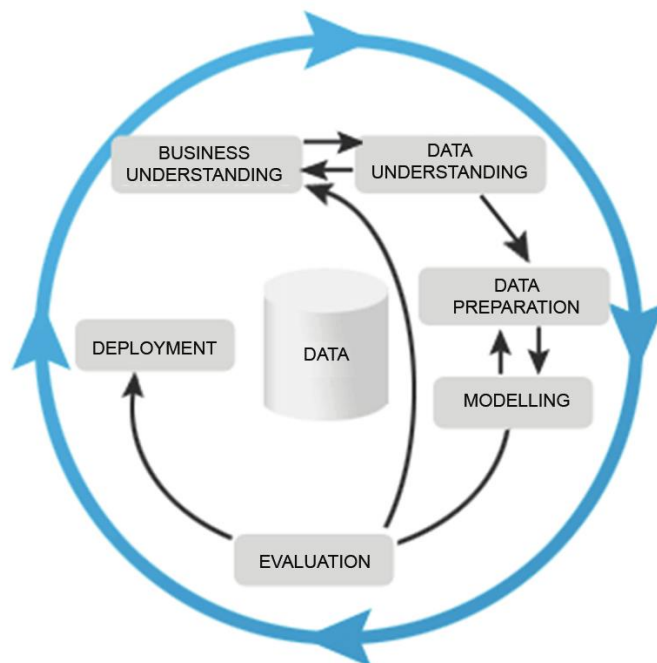


Figure 2.1 : The phases of the current CRISP-DM process model for data mining (Han & Kamber, 2006)

As it is displayed in Figure 2.1 , the current CRISP-DM process model for data mining includes the following phases:

2.1.Business Understanding

This initial phase focuses on understanding the project objectives and requirements from a business perspective and then converting this knowledge into a data mining problem definition, and a preliminary project plan designed to achieve the objectives.

2.2.Data Understanding

The data-understanding phase starts with an initial data collection and proceeds with activities in order to get familiar with the data, to identify data quality problems, to discover first insights into the data, or to detect interesting subsets to form hypotheses for hidden information.

There is a close link between Business Understanding and Data Understanding. The formulation of the data mining problem and the project plan require at least some understanding of the available data.

2.3.Data Preparation

The data preparation phase covers all activities to construct the final dataset from the initial raw data. Data preparation includes basic operations such as removal of noise if appropriate, collecting the necessary information to model or account for noise, deciding on strategies for handling missing data fields, accounting for time sequence information and known changes (Ye, 2003).

Data preparation tasks are likely to be performed multiple times, and not in any prescribed order. Tasks include tabulating, recording, and attribute selection, data cleaning, construction of new attributes, and transformation of data for modeling tools (Kantardzic, 2003).

It is widely recognized that around 80% of the resources in data mining applications are spent on data cleaning and preprocessing. The actual mining or extraction of patterns from the data requires the data to be clean since input data are the primary, if not the only, source of knowledge in these systems. Cleaning and preprocessing data involves several steps including procedures for handling incomplete, noisy, or missing data; sampling of appropriate data; feature selection; feature construction; and also formatting the data as per the representational requirements of methods (e.g., decision trees, neural networks) used to extract knowledge from these data (Han & Kamber, 2006).

To achieve an accurate data mining process, the identification of the effective features (variables) is a crucial task. In other words, recognition of the most important representative features plays a crucial role in the accuracy and speed of the data mining process. Feature selection is the problem of choosing a small subset of features that ideally is necessary to describe the target concept (L. Yu & Liu, 2003). Therefore, in the following the feature selection concept and existing algorithms have been presented.

2.3.1. Feature Selection

Feature selection is of paramount importance for any learning algorithm which when poorly done (i.e., a poor set of features is selected) may lead to problems associated with incomplete information, noisy or irrelevant features, not the best set/mix of features, among others. The ultimate objective of feature selection is to obtain a feature space with (1) low dimensionality, (2) retention of sufficient information, (3) enhancement of separability in feature space for example in different categories by removing effects due to

noisy features, and (4) comparability of features among examples in same category (Piramuthu, 2004). The optimality of a feature subset is measured by an evaluation criterion.

Feature selection algorithms designed with different evaluation criteria broadly fall into three categories: 1. Filter methods, 2. Wrapper Methods, and 3. Hybrid methods (Y.-S. Chen & Cheng, 2012; C.-L. Huang & Tsai, 2009).

- ***Filter algorithm***

The filter model relies on general characteristics of the data to evaluate and select feature subsets without involving any mining algorithm. For a given data set \mathbf{D} , the algorithm starts the search from a given subset \mathbf{S}_0 (an empty set, a full set, or any randomly selected subset) and searches through the feature space by a particular search strategy. Each generated subset \mathbf{S} is evaluated by an independent measure \mathbf{M} and compared with the previous best one.

If it is found to be better, it is regarded as the current best subset. The search iterates until a predefined stopping criterion δ is reached. The algorithm outputs the last current best subset \mathbf{S}_{best} as the final result. Compared to the wrapper and hybrid models, algorithms of the filter model are independent of any learning model, therefore do not have bias associated with any learning models. Another advantage of the filter model is that it allows the algorithms to have a very simple structure, which usually employs a straightforward search strategy, such as backward elimination or forward selection, and a feature evaluation criterion designed according to particular criterion. The benefit of the simple structure is two-fold. First, it is easy to design, and after it is implemented, it is easy to understand for other researchers. This explains why most feature selection algorithms are of the filter model. Also, in real-world applications, many most frequently used feature selection algorithms are also filters. Second, since the structure of the algorithms is simple, they are usually swift.

- ***Wrapper algorithm***

Feature selection algorithms of Wrapper model require a predetermined learning algorithm and use its performance on the provided features in the evaluation step to identify the relevant feature. In the wrapper method, the goal is to find a subset of size r from n variables ($r < n$) that maximizes the predictor performance (S. n. Maldonado & R. Weber, 2009). The method utilizes the learning mechanism as the fitness function and seeks the best subset of the features while standard optimization techniques with learning mechanisms for ranking of the subsets are possible. Kohavi and John (1997) have a leading role in the popularization of the wrapper approach that is powerful in feature selection, but it has its computational complexities, and it is more time consuming than Filter method (C. F. Huang, 2012). For a given dataset of \mathbf{G} with N features, the Wrapper approach starts from a subset of \mathbf{F}_0 (an empty set, a full set, or any randomly selected set) and with a particular strategy searches the features space. It evaluates each generated subset of

F_i by applying a learning model that considers F_i as an input and if the learning model performance improves with F_i ; F_i is regarded as the best current subset. Then the wrapper modifies F_i by adding or eliminating features and performing the search until coming to the predefined stopping criterion (Kabir & Islam, 2010). The above-mentioned feature searching problem is NP-hard, and the number of local minima can be quite large and naturally a wide range of heuristic search strategies including forward selection, backward elimination, hill-climbing, branch, and bound algorithms, and metaheuristics algorithms like simulated annealing and genetic algorithms (GAs) have been used (C.-F. Huang, 2012).

- **Hybrid algorithm**

The hybrid model attempts to take advantages of the two models by exploiting their different evaluation criteria in different search stages. The hybrid model is recently proposed to handle large data sets. It makes use of both an independent measure and a mining algorithm to evaluate feature subsets: it uses the independent measure to decide the best subsets for a given cardinality and uses the mining algorithm to select the final best subset among the best subsets across different cardinalities. Comparing with the wrapper model, hybrid algorithms are usually more efficient since they look into the structure of the involved learning model and use its properties to guide feature evaluation and search. In recent years, the embedded model is gaining increasing interests in feature selection research due to its superior performance. Currently, most embedded feature selection algorithms are designed by applying L_0 norm (Guyon & Elisseeff, 2003; H Liu & Setiono, 1996) or L_1 norm (Huan Liu, Motoda, Setiono, & Zhao; Yunyue Zhu & Shasha, 2003) as a constraint to existing learning models to achieve a sparse solution. Generally, compared to the filter model, feature selection algorithms of the wrapper and embedded models can usually select features that result in higher learning performance for a particular learning model, which is used in the feature selection process.

2.4. Modeling

Having the data mining model and task defined, next would be the data mining methods to build the approach based on discipline involved. There is a close link between Data Preparation and Modeling. The methods used for data mining are classification, clustering, and forecasting. Here a brief description of these three techniques is described as follows.

2.4.1. Classification and forecasting methods

Classification is one form of data analysis that can be used to extract models describing important data, which predicts categorical (discrete, unordered) labels. Data classification is a two-step process; in the first step, a classifier is built describing a predetermined set of data classes or concepts. This is the learning step (or training phase), where a classification algorithm builds the classifier by analyzing or “learning from” a training set made up of database tuples and their associated class labels. Initially, training data are analyzed

by a classification algorithm; then the test data are used to estimate the accuracy of the classification rules. If the accuracy is considered acceptable, the rules can be applied to the classification of new data tuples. Classification is one form of prediction, where the value to be predicted is output class labels. In another word, predicting class labels is classification, while if we have to predict the values of output, then the problem changes to the forecasting/regression problem. In numerical prediction (often called regression) one wish to predict a numerical value, such as a company's profits or a share price. Classification and forecasting have numerous applications, including fraud detection, target marketing, performance prediction, manufacturing, medical diagnosis and stocks return or stock risks and many other applications. For example, one can build a classification model to categorize the stock return as either high, average or low.

Classification methods: common models being used for classification problems are Artificial Neural Networks, Decision Trees, Nearest Neighbor Matching, Association Rules, and Bayesian Classification.

- **Artificial neural networks**

Artificial neural networks are used to map an input to the desired output with a nonlinear mathematical function approximation. It is inspired by the behavior of neurons located in the brain (Priddy & Keller, 2005). Neural networks are non-parametric estimators that can be used for several kinds of tasks such as forecasting, clustering, and optimization (Mehrotra, Mohan, & Ranka, 1997).

The inputs of a neural network are vectors of variables corresponding to an observation. These vectors are weighted and combined by linear filters and become the inputs of hidden layers where non-linear computation is done on inputs. Network output will be calculated by a function called activation function that receives the output of hidden layers and calculate the output of the network (Gershenfeld, 1999).

In the following the mathematical explanation of multilayer perception (G. Zhang, Patuwo, & Hu, 1998) which is one of the most favorite models of Artificial Neural Network (ANN) has been presented (See Eq. 2-1):

$$F_t = \beta_0 + \sum_{n=1}^N \beta_n W(\theta_{0i} + \sum_{j=1}^m \theta_{nj} x_j) \quad \text{Eq. 2-1}$$

where m is the number of inputs (x_j) and N is the number of nodes in the hidden layer. θ_{nj} is the weight for inputs and β_n is the weight for the hidden layers. Here the zeroth indices in the weights refer to the bias nodes in each layer. One can use either sigmoid or hyperbolic tangent functions as the transfer function W . The whole process of learning is achieved by adjusting the weight parameters and the network is updated each time it has been feed by new data. After the parameters are updated, the desired outcome will be the classification of the data. More on optimization methods can be found in (Ragg, Braun, & Landsberg, 1998).

- **Decision Tree**

Decision tree algorithm is a non-parametric and non-linear machine learning technique. This technique takes advantage of a hierarchical structure for recursively segmenting training data, and therefore, it has great flexibility and interpretability in data analysis. The most common strategy to induct a decision tree is greedy top-down construction, which recursively partitions the data into subsets until the stopping criterion has met. The stopping criterion is crucial so that it can prevent growing branches that do not affect the tree quality (Murthy, 1998).

Some of these stopping rules are:

- 1- The number of observation in a node is less than a pre-specified threshold.
- 2- All observation assigned to a node belongs to the same class.
- 3- Depth of the node is more than some pre-specified limit.
- 4- Nodes purity are more than a specified threshold (Esposito, Malerba, Semeraro, & Kay, 1997).

Decision trees are often prone to over-fitting according to high variance. Hence, methods are proposed to find a right-sized tree. The most popular method is pruning trees (Rizzo, d'Amato, Fanizzi, & Esposito, 2017), so in order to have a high-quality tree, first the complete tree is built and then inefficient sub-trees, that do not influence cost function (accuracy) significantly, are removed.

The evaluation function used for splitting in classification trees in CART (classification and regression tree) method is Gini index, which describes the chance of coming up with a false node for the data if the node was chosen randomly from the nodes' distribution. The Gini index, which measures the impurity of the node, can be stated as

$$Gini(t) = 1 - \sum_j [P(j|t)]^2 \quad \text{Eq. 2-2}$$

Where $P(j|t)$ is the proportion of finding the data class j in node t (node purity). The aim is to minimize the Gini index. From the formula, it can be inferred that if the classification is done in a perfect way, the Gini index would be zero (Friedman, Hastie, & Tibshirani, 2001; Kumar, 2013).

- **Nearest Neighbor Matching**

This method relies on identifying (say) the five examples that are closest in some sense to an unclassified one. If the five 'nearest neighbors' have grades Second, First, Second, Second and Second, one might reasonably conclude that the new instance should be classified as Second (Han & Kamber, 2006).

- **Association Rules**

Sometimes researchers wish to use a training set to find any relationship that exists amongst the values of variables, generally in the form of rules known as association rules. There are many possible association rules derivable from any given dataset; most of them are little or with no value. Therefore, it is usual for association rules to be stated with some additional information indicating how reliable they are, for example, the following is an association rule mined from a data set shown with its confidence and support (Han & Kamber, 2006), Where “^” represents a logical “and”

$$Age = (youth \wedge credit = OK) \text{ buys computer} = yes [support = 20\%, confidence = 93\%]$$

- **Bayesian Classification**

Bayesian classifiers are statistical classifiers. They can predict class membership probabilities, such as the probability that a given tuple belongs to a particular class. Bayesian classification is based on Bayes' theorem (Han & Kamber, 2006).

2.4.2. Clustering

Clustering is the task of segmenting a heterogeneous population into some more homogenous clusters (Khan & Ahmad, 2004; Linoff & Berry). It is different from classification in that clusters are unknown at the time the algorithm starts. In other words, there are no predefined clusters. A cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters. A cluster of data objects can be treated collectively as one group and so may be considered as a form of data compression. Common tools for clustering include Partitioning methods, Hierarchical methods, Density-based methods, Grid-based methods, and Model-based methods.

- **Hierarchical methods**

A hierarchical method creates a hierarchical decomposition of the given set of data objects. A hierarchical method can be classified as being either agglomerative or divisive, based on how the hierarchical decomposition is formed. The agglomerative approach, also called the bottom-up approach, starts with each object forming a separate group. It successively merges the objects or groups that are close to one another, until all of the groups are merged into one (the top most level of the hierarchy), or until a termination condition holds. The divisive approach, also called the top-down approach, starts with all of the objects in the same cluster. In each successive iteration, a cluster is split up into smaller clusters, until eventually, each object is in one cluster, or until a termination condition holds.

2.5.Evaluation

At this stage in the project, you have built one or more models that appear to have high quality, from a data analysis perspective. Before proceeding to final deployment of the model, it is important to more thoroughly evaluate the model, and review the steps executed to construct the model, to be certain it properly achieves

the business objectives. A key objective is to determine if there is some important business issue that has not been sufficiently considered. At the end of this phase, a decision on the use of the data mining results should be reached. In other words, in this step, we will choose the appropriate forecasting model to be employed. To do so, we compare the result of a specific error measure on different candidate models and select the model with the lowest error. There are different kinds of error measures. The error measurements have been joined in the seven groups: absolute forecasting errors, measures based on percentage errors, symmetric errors, measures based on relative errors, scaled errors, relative measures, and other error measures.

2.6. Deployment

Creation of the model is not the end of the project. Even if the aim of the modeling is information acquisition from the data, the gained information will need to be organized and presented in an applicable and appropriate way. Deployment is the process of using your new insights to make improvements. In deployment, the results of the process which are confirmed to work are utilized. Depending on the situation, the deployment phase can be as straightforward as generating a simple report or be complicated, such as implementing a repeatable data mining process. In this thesis, the process of each model is presented with flowcharts and figures with a detailed explanation of each step. Besides, each model is implemented for a specific data set, and the results of methods are provided. Thus, it will be clear and comprehensive what actions will need to be done to make the models applicable to other cases.

Chapter 3

Developing an approach to evaluate stocks by forecasting effective features with data mining methods

Summary of the third chapter

In this chapter, a novel approach is developed to predict stocks return and risks. In this three-stage method, first, through a comprehensive investigation, all possible features which can be effective on stocks risk and return are identified. Then, in the next stage, risk and return are predicted by applying data mining techniques for the given features. Finally, we develop a hybrid algorithm, based on a filter and function-based clustering; the important features in risk and return prediction are selected then risk and return re-predicted. The results show that the proposed hybrid model is a proper tool for effective feature selection, and these features are good indicators for the prediction of risk and return. To illustrate the approach as well as to train data and test, we apply it to the Tehran Stock Exchange (TSE) data from 2002 to 2011.

3.1.Introduction

A reliable prediction of the company's financial status provides a situation for the investor to more confident investments and gaining more profits (C. F. Huang, 2012). One can refer to different studies about share gaining and return prediction, for example, time series stock price prediction model (Araújo & Ferreira, 2013), Buy-hold-sell prediction model (Wu, et al., 2014; X. Zhang, Hu, Xie, Zhang, et al., 2014), Index prediction model with ANFIS (Svalina, et al., 2013) and so on. However, unlike the return, risk has been rarely considered for prediction, while customers usually balance their return for a proper level of risk, then clearly both risk and return are important factors in financial decision making (Barak, Abessi, & Modarres, 2013; Tsai, Lin, Yen, & Chen, 2011). Without risk evaluation, the portfolio efficient frontier does not make sense. Thus, this chapter implements the forecasting of both risk and return of stocks, which has a tremendous effect on price setting. Also, up-down prediction of stock movement such as (Patel, Shah, Thakkar, & Kotecha, 2014; H. Yu, Chen, & Zhang, 2014; X. Zhang, Hu, Xie, Wang, et al., 2014) cannot result in precision view of stock future and investors gain. While classifying the amount of risk and return to different categories like our method gives more specific and explicit knowledge.

Therefore, in this chapter, the prediction of risk and return classes with different classification algorithms are investigated.

To predict risk and return variables accurately, the effective factors need to be identified. One of the key issues of stock prediction design lies on how to select representative features for prediction(X. Zhang, Hu, Xie, Wang, et al., 2014).

Most studies in this area focus on technical features, financial ratios, or macroeconomic indicators. For example, Tsai and Hsiao (2010) studied eight financial ratios and 16 macroeconomic indicators as the main features to predict stock return by Back Propagation in the Taiwan stock market. J.-H. Cheng, Chen, and Lin (2010a) conducted a comprehensive study on macroeconomic and technical features and studied eight financial ratios and ten macroeconomic indicators to investigate their effect on return variation in the Taiwan stock market. By applying probabilistic back propagation algorithm (PBP), Rough set, and C4.5 Tree², they achieved 76% accuracy. F. A. de Oliveira, Nobre, and Zárate (2013) use 15 technical indicators and 11 fundamental indexes to the prediction of stocks movement in Petrobras with artificial neural networks and obtain 87.50% for direct prediction. Tsai, et al. (2011) considered 19 financial ratios and 11 macroeconomic indicators in Taiwan stock market by combining logistic regression Algorithm, MLP Back Propagation, and CART Tree to investigate their effect urn (negative or positive) on the stock return and achieved 66.67% accuracy based on Bagging and Voting algorithms. In the majority of studies, as mentioned, the focus is mostly on financial ratios, macroeconomic indicators, and technical indicators based on experts' ideas to predict returns. However, this chapter presents a systematic and efficient methodology for comprehensive searching the potential representative features on the stock market in 3 categories of a financial ratio, profit & loss reports, and Stock pricing models and not arbitrarily choosing likely effective features.

Furthermore, many studies have claimed and verified that feature selection (FS) is the key process in stock prediction modeling (Tsai & Hsiao, 2010). Therefore, in this chapter, a novel hybrid feature selection algorithm based on a filter and function-based clustering method is applied to select the important features. What makes the proposed approach different from the previous ones is that the combination of 9 different feature selection algorithms is considered with a function-based clustering algorithm. The hybrid model of this chapter employs the power and advantage of correlation based algorithms like Chi-Square, One-R in addition to the power of classified errors based, interval based, and information based algorithms like Support Vector Machine (SVM), Relief-f, and Gini index/ gain ration algorithms respectively. The effectiveness of our model is illustrated with the prediction of both risk and return of stocks and then analyzing the results with and without implementing our hybrid feature selection algorithms.

To sum up, in the first stage of the chapter, a complete list of likely effective features on the stocks risks and returns are identified. After developing an appropriate database in the second stage, different classification

² Additional information about these algorithms is provided in Appendix A.

algorithms are used to predict the risk and return. We also scrutinize on the effect of their results to our database based on the feature-oriented viewpoint. Finally, in the third stage, a novel hybrid feature selection algorithm based on the filter and function-based clustering method is applied to select the important features which affect the prediction of risk and return.

The contribution of this chapter is summarized as follow:

- A comprehensive and systematic study to identify the likely effective features in risk and return prediction.
- Stock risks as well as return prediction with different classification methods.
- Designing a hybrid feature selection algorithm based on the filter and function-based clustering.
- Finally, each algorithm with a feature-oriented viewpoint is analyzed. The results indicate the factors which cause strength and weakness of that algorithm. As a result, the nature of each feature is provided according to the amount of interference variable in their prediction.

The rest of the chapter is organized as follows. In Section 3.2, the proposed model is presented, which has three stages. In Section 3.3, the approach is implemented for real sample data from the Tehran Stock Exchange (TSE). The results are analyzed in which the predictions with and without considering important effective features are also compared. Then in Section 3.4, a discussion on real return and risk prediction with important features has been represented. Finally, some conclusion and future research directions are provided in Section 3.5.

3.2.Proposed model

The proposed algorithm, which consists of three stages, is shown in Figure 3.1. In the first stage, a database is developed, and data is pre-processed. Risk, as well as real return, is predicted with classification algorithms in the next stage. A hybrid feature selection algorithm is also presented in the third stage and risk and return are re-predicted based on selected features.

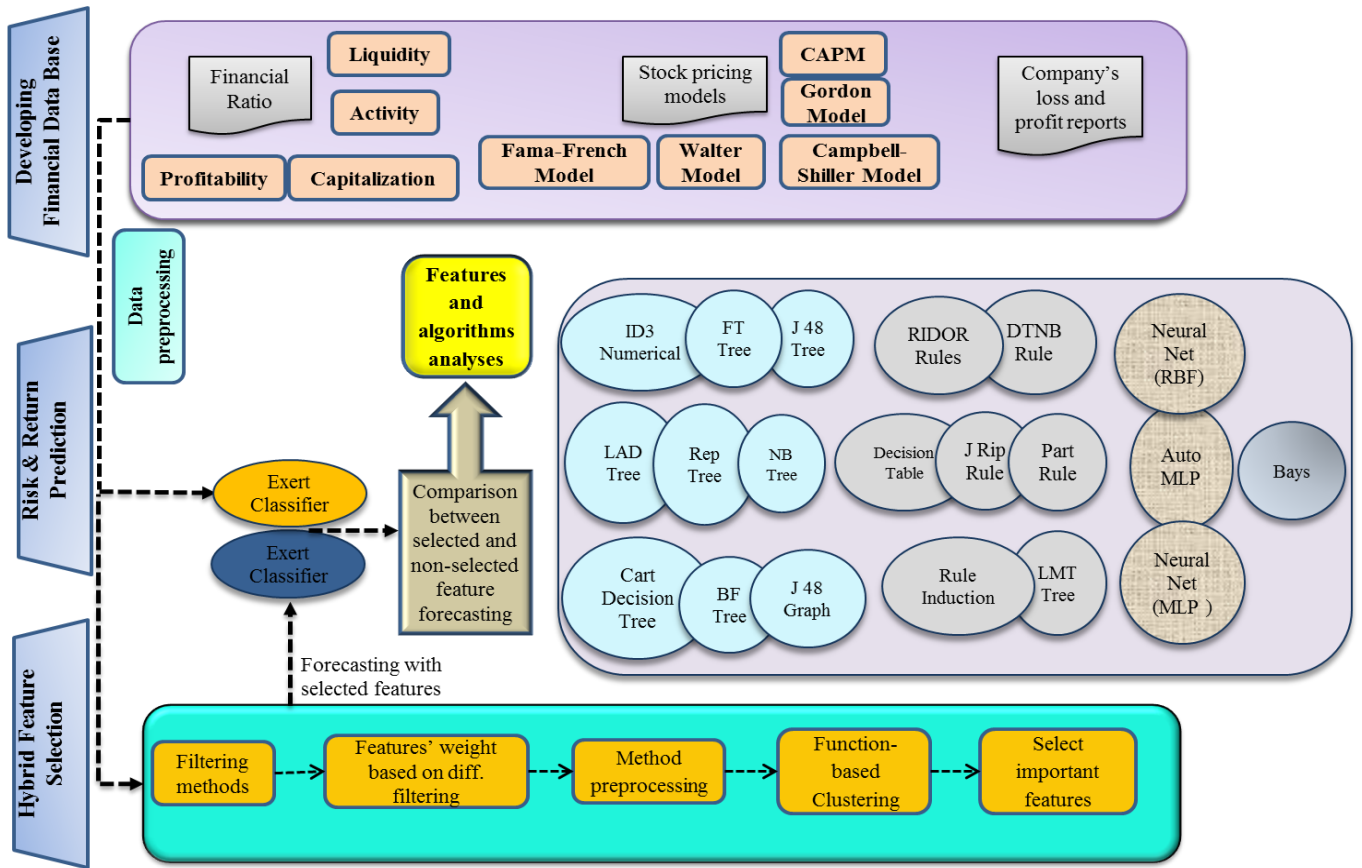


Figure 3.1 Conceptual design of the proposed model

3.2.1. First stage: Developing financial database

This stage we utilize the concepts and techniques of input features, response variables, and preprocessing models.

• Input features

First, we analyze and gather important features from the company's financial ratios and the profit & loss reports, as well as stock pricing models (Table 3-1).

- **Financial Ratio:** To have a complete list of effective features we gather four general groups of financial ratio as a part of the input variables of companies' database. The importance of these features is discussed in many studies [see (Bauer, Guenster, & Otten, 2004; Bernstein & Wild, 1999; Carnes & College, 2006; C.-F. Huang, 2012; Omran & Ragab, 2004; Sadka & Sadka, 2009; Soliman, 2008)], also see financial ratio's part of Table 3-1.
- **Stock pricing models:** We review different stock pricing models (Capital Asset Pricing Model (CAPM), Gordon, Walter, Campbell-Shiller, and Fama-French) and obtain other important factors which effective on the risk and return prediction of stocks, see Table 3-2 [(Kaplan & Ruback, 1995)(Brealey, Myers, & Allen, 2007; Fama & French, 1993, 2012; Gordon, 1982; Hjalmarsson, 2010; W.-S. Lee, Tzeng, Guan, Chien, & Huang, 2009; Lewellen, 2004; Mukherji, Dhatt, & Kim, 1997)]

- Company's profit and loss reports: By using the profit and loss reports of companies, the other added factors are extracted. In Table 3-1, all input variables of the financial model are provided.

Table 3-1 Stock pricing models

CAPM	$R = r_f + \beta (r_m - r_f), (1)$	This model explains the connection between expected return and risk and it is used for bonds pricing with risk.(Kaplan & Ruback, 1995) R: expected Return, r_f : rate of return without risk, β : systematic risk, r_m : market expected return. A brief description about this formula can be found in Appendix A. On the basis of this model, r_m, r_f, β are added to database.
Gordon Model	$P = \frac{DPS}{k - g}, (2)$ $DPS = EPS \times DPR$	Gordon has suggested this model using the investment of retained earnings to stock pricing(Gordon, 1982). g: stock profit increase, K: shareholder's expected return ratio. From this model two important factors EPS and DPS are achieved. ROE feature have been mentioned also in financial ration before. In addition to this, four other features that were obtained from EPS have been regarded and inserted into the model as follows: EPS prediction of companies in fiscal year, EPS coverage, prediction difference percentage of EPS with the real amount, and EPS growth ration in compare to the previous fiscal year(Hjalmarsson, 2010). Gordon Model is used in different capital market discussions like(W.-S. Lee, et al., 2009).
Walter Model	$P = \frac{DPS + (EPS - DPS) r/k}{k}, (3)$	P: stock market price of each stock, r: internal rate of return, EPS-DPS: cumulative profit per share, K: capitalrate cost(Brealey, et al., 2007). According to this model, cumulative profit per share is known as a criterionin the database.
Campbell - Shiller Model	P/E , P/S ratio	This model calculatesthe stock P/E average by using Market data (Brealey, et al., 2007). From the literature, it was clarified that P/E parameter is very important for analyzing and predicting the stock price, and it is inserted to database(Hjalmarsson, 2010; Lewellen, 2004). In addition to this, P/S ratio that is result of Stock pricedivided to each stock sale is also inserted to database(Mukherji, et al., 1997).
Fama-French Model	$r_i = \alpha + \beta (r_m) + \beta_{size}(Size) + \beta_{\frac{B}{M}}(\frac{B}{M}), (4)$	Fama- French offer β , size and book value to market value's Model with the help of CAPM Model as a multivariate regression to study the Factors affecting portfolio returns(Fama & French, 1993, 2012) The first part of the model is similar to Sharp model. The second part shows the company size which is a factor showing the company's capital and third part indicates the book value to market value. By using this model, company's capital, stock Book value and stock Market value are inserted to the database as 3 important factors. The other models like Glassman-Host and kernel are derived from the introduced methods and they do not help this research in finding new features.

- Company's profit and loss reports: By using the profit and loss reports of companies, the other added factors are extracted. In Table 3-2, all input variables of the financial model are provided. In Appendix C detailed information about input features are provided.

Table 3-2 Financial input features

Category		Features
Financial ratio	Liquidity ratios	current ratio , quick ratio, current assets ratio, Net working capital, Liquidity ratios
	Activity ratio	Average payment period, current assets turn over, Fixed asset turnover, Total asset turnover
	Capitalization ratio	Equity ratio, Debt coverage ratio, debt to total assets ratio, debt to equity ratio, long-term debt to equity ratio, current debt to equity ratio.
	profitability ratio	Percentage of net profit to sale, Percentage of Operating profit to sale, Percentage of Gross profit to sale, Percentage of net profit to Gross profit, Return on asset (after tax) ROA, Return on equity (after tax) ROE, working capital return percentage, fixed assets return percentage, assess the loan usefulness
Stock pricing models	Capital Asset Pricing Model	r =return ration without risk β = stock beta coefficient (systematic risk) r_m =expected return from market
	Gordon Model	EPS, DPS, EPS prediction, EPS cover, prediction difference percentage of EPS with the real amount, EPS growth ration in compare to the previous fiscal year.
	Campbell - Shiller Model	P/E, P/S
	Walter model	Stock cumulative profit
	Fama- French Model	Company's capital(investment), stock book value, stock market value
Company's loss and profit reports	Total predicted income (last income prediction in the current fiscal year), Total income growth % (total real income / (total real income - total predicted income)), predicted Profit margins (last profit ratio / company's income in the current fiscal year), profit margin growth rate (real profit margin / (real profit margin – predicted profit margin)) and Efficiency (Percent of daily trading volume / company's daily value in the before period).	

• Response variables

The most important response variables in the model are the real return and the risk, as follows:

$$R = \sqrt[n]{\left(1 + \frac{r_1}{100}\right) \left(1 + \frac{r_2}{100}\right) \dots \left(1 + \frac{r_n}{100}\right)} \quad \text{Eq. 3-1}$$

where $r_1 \dots r_n$ = return of $1 \dots n^{\text{th}}$ periods. So, the quarterly return of each company as described in the Eq. 3-1 will be the geometric average of its past quarters' return.

The risk is defined as the standard deviation of the stock return, as follows.

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=0}^n (r_i - E(r))^2} \quad \text{Eq. 3-2}$$

In the forecasting model, real return and risk of the next time period are considered as target variables, are predicted by input features.

• Data pre-processing

Data preparing stage is an important part of the approach. Furthermore, it is time-consuming in the data mining process, described as follows.

- Removing high correlation features: Features with higher than a predefined correlations percent based on Pearson test are removed.

- Missing data: Defected records caused by incomplete information of the company or the company's negligence in reporting are also deleted from the database. Some decision tree algorithms and K nearest neighbor techniques do not need to replace the missing data.
- Finding outlier data: To find outlier data in database, we use the distance-based approach which is based on data intervals (Knorr & Ng, 1999), density approach (Breunig, Kriegel, Ng, & Sande, 2000) in which a parameter named Local Outlier Factor (LOF) is specialized to each sample based on K-Nearest neighbor density. Samples with high LOF are known as Outlier points, clustering approach (Hong & Wu, 2011) within the use of k means clustering algorithm, and deviation method (Hong & Wu, 2011).
- Testing stationarity of data: in the training machine learning models, it is important to be sure that distribution of training set is similar to the test set which makes the models applicable for the future horizon of the time series. This infers that the time series data set should be stationary. Thus, before constructing the models, the time series will be tested whether they behave as a stationary time series or not. Typical tests for detecting stationarity of time series which also known as unit root tests are the Kwiatkowski-Phillips-SchmidtShin test (KPSS) (Kwiatkowski, Phillips, Schmidt, & Shin, 1992), augmented Dickey-Fuller test (ADF), Philips-Perron test (PP) (Perron, 1989). To be ensure about the stationary concern, all of these three tests are applied.

3.2.2. Second stage: Risk and return prediction with classification methods

Generally, researchers and scholars are seeking to achieve a more scientific model, ranging from Portfolio Theory by Markowitz in 1952 and Sharp assets pricing models in 1964, to Fama- French in 1992. However, they cannot solely evaluate price, risk, and return well. Bartholdy and Peare (2005) compared the CAPM and Fama-French Model while it appears that the latter can better explain the return deviation and can give better evidence. However, regarding the real data, none of them can explain return well. Q. Cao, Leggio, and Schniederjans (2005) concluded that the neural network is much more powerful than the Fama- French Model in stock return prediction. Dastgir and Afshari (2004) compared Walter, Gordon and the current value of future cash flow stock pricing models in the Tehran Stock Exchange and observed that real prices and prices obtained by models were not equal. As these studies show the traditional methods cannot necessarily estimate accurately. Thus, it is necessary to apply some methods to be able to determine the complexity of the data. Some researchers have used different methods, like neural networks and statistical methods. Among these results, the conclusion gained by machine learning algorithm and data mining are prominent (Patel, et al., 2014).

Phichhang Ou and Hengshan Wang (2009) and Lai, Fan, Huang, and Chang (2009) concluded that decision tree methods have outstanding performance in stock return prediction. Also, what is important is the rules obtained from the rule-based algorithms and trees, since these rules conduct investors to build up their buy and sell strategies of trading.

On the other hand, the output of the methods that are applied in this area (like SVM³ and NN⁴) which do not use rules for prediction is not appropriate for practitioners. The decision tree structure is more comprehensive, transparent, and rational. Based on what was discussed, this chapter focuses on the tree and rule-based algorithms in order to be more appropriate for investors and analysts. Levin and Zahavi (2001) concluded that the data correlation problem in tree algorithms is more transparent than statistical algorithms, and it can be solved by Pruning algorithms. T.-S. Chang (2011) compared CART⁵, Back Propagation, and CART-Back Propagation hybrid method from stock price prediction based on fundamental data and concluded that Back Propagation and Decision Tree accuracy perform better than the Hybrid methods.

In this chapter, by using different classification methods, risk and return are predicted based on the given features and database. A comparison between different methods is performed. This section is done for two times. In the first time, the prediction is made with all features, but in the second time, the best selected features from the hybrid feature selection algorithm are predicted. A comprehensive comparison between these two predictions is also done. In other words, in this chapter, we compare the accuracy of risk and return forecasts with and without feature selection, based on different classification methods and explain the effect of feature selection on classification methods. The classification algorithms are shown in Figure 3.1.

Testing Strategy

In order to get robustness prediction, we perform a 10- fold cross-validation model on the predictors (duda, Hart, & Strok, 2001). This method has been proved to be statistically good enough in evaluating the performance of the predictive model (Mitchell, 1997). In 10-fold cross-validation, the training set is equally divided into 10 different subsets. Nine out of 10 of the subsets are used to train the classifier, and the tenth subset is used as the test set. The procedure is repeated ten times, with a different subset being used as the test set, and the best result has been chosen.

In order to reliably evaluate the predictors, we consider not only prediction accuracy but sensitivity and specificity. The accuracy of a predictor on a given test set is the percentage of test set tuples that are correctly predicted by the predictor. Sensitivity is also referred to the proportion of positive tuples that are correctly identified while specificity is the proportion of negative tuples that are correctly identified (Han & Kamber, 2006). Based on true and predicted conditions which are shown Table 3.3 accuracy, sensitivity, and specificity are defined as the following:

$$\text{Accuracy} = \frac{\text{True positive} + \text{True negative}}{\text{Total population}} \quad \text{Eq. 3-3}$$

$$\text{Sensitivity} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \quad \text{Eq. 3-4}$$

³ Support Vector Machine

⁴ Neural Network

⁵ Classification And Regression Tree

$$\text{Specificity} = \frac{\text{True Negative}}{\text{True Negative} + \text{False Positive}}$$

Eq. 3-5

Table 3-3 Confusion matrix

		True condition	
		Condition positive	Condition negative
Predicted condition	Predicted condition positive	True positive	False positive, (Type I error)
	Predicted condition negative	False negative, (Type II error)	True negative

Prediction accuracy for five classes can be measured by a confusion matrix shown in Table 3-4 with the Eq. 3-6.

Table 3-4 Confusion matrix for five classes

		Predicted class				
		Very low	low	normal	high	very high
Actual class	Very low	a ₁	b ₁	c ₁	d ₁	e ₁
	Low	a ₂	b ₂	c ₂	d ₂	e ₂
	Normal	a ₃	b ₃	c ₃	d ₃	e ₃
	High	a ₄	b ₄	c ₄	d ₄	e ₄
	Very high	a ₅	b ₅	c ₅	d ₅	e ₅

$$\text{Accuracy} = \frac{a_1 + b_2 + c_3 + d_4 + e_5}{\sum_{i=1}^5 a_i + b_i + c_i + d_i + e_i}$$

Eq. 3-6

3.2.3. Third stage: Hybrid feature selection

Based on particular conditions in the stock exchange, occasionally we encounter many attributes whereas some of them no longer have useful information and complicate the condition. For this reason, feature selection is one of the very crucial aspects that has a highly regarded recommendation (C.-F. Huang, 2012; C.-J. Huang, Yang, & Chuang, 2008; Tsai & Hsiao, 2010).

In this section to investigate the features which have greater effect on risk and return and better analysis of algorithms results a novel feature selecting method in 2 levels is established. It should be noted that feature selecting in capital markets issues has double importance. The reason is that we encounter with so many

features that are either useless or have low information value. Thus, dealing with these features is time wasting without any gain. Feature selection methods are generally divided into three categories: 1. Filter methods, 2. Wrapper Methods, and 3. Hybrid methods(Y.-S. Chen & Cheng, 2012). In this approach, we use a hybrid model based on the combination of filter and function-based clustering methods to extract a set of efficient features as a follow:

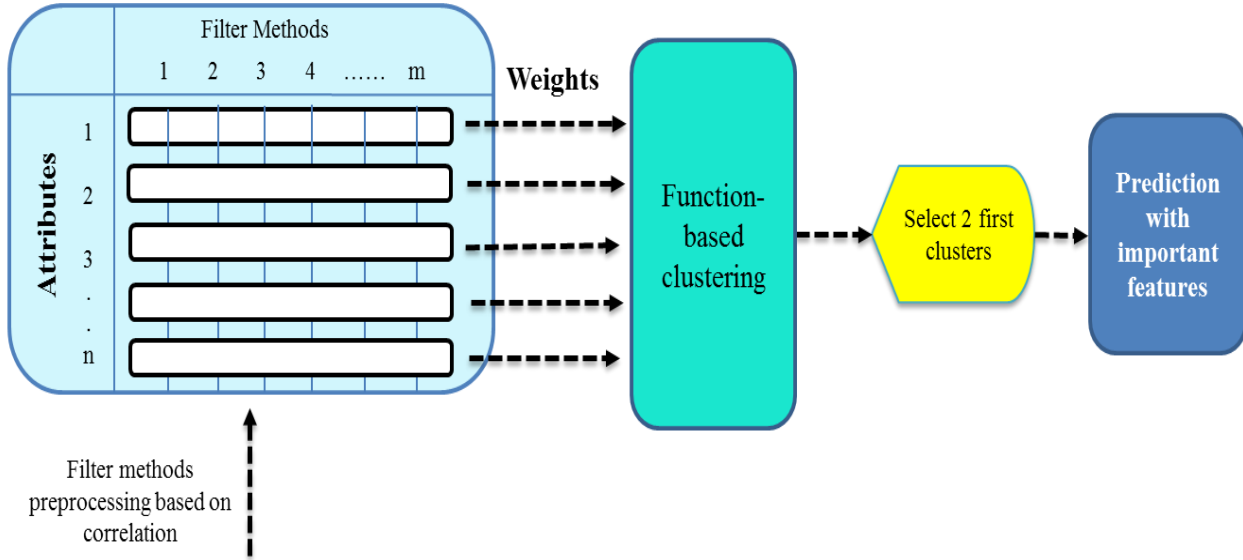


Figure 3.2 Hybrid feature selection

- **Filter methods**

According to Witten and Frank (2011) seven algorithms were defined as Filter method: Chi-square (Kononenko, 1994), info gain (Dumais, Platt, Heckerman, & Sahami, 1998), Gain Ratio (duda, et al., 2001), Relief-f (Kononenko, 1994), Consistency (H Liu & Setiono, 1996), One R (Holte, 1993) and CFS (Mark Hall, 1998). Also, Symmetrical Uncertainty and SVM algorithm are also used for weighting the features(Y.-S. Chen & Cheng, 2012). In this section, to compare the importance of each feature using mentioned methods, a comprehensive analysis was conducted on the features, and eventually, the weightings of features are presented.

- **Function-based clustering method**

After attaining the weights of the features by different filter based algorithms, we have attributes with m attributes' weigh, and then we need a model to determine the important features' clustering between these weighted attributes. In this section, we use Li (2006a) function-based clustering method. This model is based on hierarchical divisive clustering method which begins with one cluster including all objects, $(X_{n \times m})$. For the object x_1, \dots, x_n , we denote the vector of the group membership of objects as $z = (z_1, \dots, z_n)^T$, where $z \in Z$, and Z is the space of sign vectors defined to be

$$Z = \{z = (z_1, \dots, z_n)^T | z_i = \pm 1\} \quad \text{Eq. 3-7}$$

All objects that are associated with an entry of 1 in z are classified into one group, whereas the others with an entry of -1 are classified into the other group.

Then by using the model of multivariate analysis of variance defined to be

$$x = \mu + z_i\gamma + \varepsilon_i, i = 1, 2, \dots, n \quad \text{Eq. 3-8}$$

where the error vectors ε_i are assumed to be normally distributed with a zero mean and a common covariance matrix V , i.e. $N(0, V)$. Also ε_i and ε_j ($i \neq j$) are assumed to be independent. Then by maximum likelihood, the clustering problem is formulated as a least squares optimization problem.

$$\min_{\alpha, \beta, z \in Z} \{(z - \alpha - X\beta)^T(z - \alpha - X\beta)\} \quad \text{Eq. 3-9}$$

Simultaneously the unknown vector of cluster membership and the coefficients of the linear clustering function are estimated. The computation of the clustering-function-based method will be converted to that of sign analysis(Li, 2006b), and by problem-solving, two clusters is achieved.

Next, one of these groups based on higher within-group dispersion matrix is further divided into two dissimilar subgroups. The process continues until some stopping criterion has been satisfied. Most of the stopping criteria are based on within-group dispersion and/or between-group dispersion matrices.

By this approach, we use the advantages of different filter methods and use these weighting attributes by function-based clustering method to make a more accurate decision of effective feature.

3.3.Experimental results and analysis

In this chapter, a database including 44 input features and 2 goal features are gathered from TSE data from 2003 to 2012. The resulting database has 1963 records for 400 companies. Because of the quarterly availability of fundamental feature data for stocks in the TSE, all data including input features and target values are in the quarterly time period. In this chapter, real return and the risk which are the next quarter risk and return respectively, are predicted by the feature values of past available quarters.

According to a group of experts, 5 intervals were introduced for the real return: very high with a range higher than 9.3 , high with the range of 4 to 9.3 , average with a range of 1.14 to 4, low with the range of -1.3 to 1.14 and very low that lower than -1.3 . These intervals are proposed by market professionals based on the historical performance of the market and alternative investment opportunities which were available for investors. Risk is also classified in 3 intervals: high in the range of higher than 15.5, average in the range of 6.3 to 15.5 and low in the range of lower than 6.3. According to literature, it is found out that to predict return, negative and positive return (Tsai, et al., 2011; J.-L. Wang & Chan, 2006) and negative and positive return trend (Enke & Thawornwong, 2005) are used [see also (Patel, et al., 2014; H. Yu, et al., 2014; X. Zhang, Hu, Xie, Wang, et al., 2014)]. For more accuracy, we increased the prediction intervals. These intervals give more information to investors, and they can develop a balance between the share price and the future gained

return. The information which is limited to the company's profitability or losses does not help them very much. Besides this, risk has been rarely mentioned in the prediction field of the stock exchange. We can conclude whether the proposed return range is optimal or not just by knowing the risk amount. The previous studies of the field have just focused on return prediction while these two features together show the Portfolio Efficient Frontier and investors can use it to select the optimal portfolio. However, it is important to note that we did not predict the risk and return simultaneously, and each output considered separately. The process is illustrated in Figure 3.3.

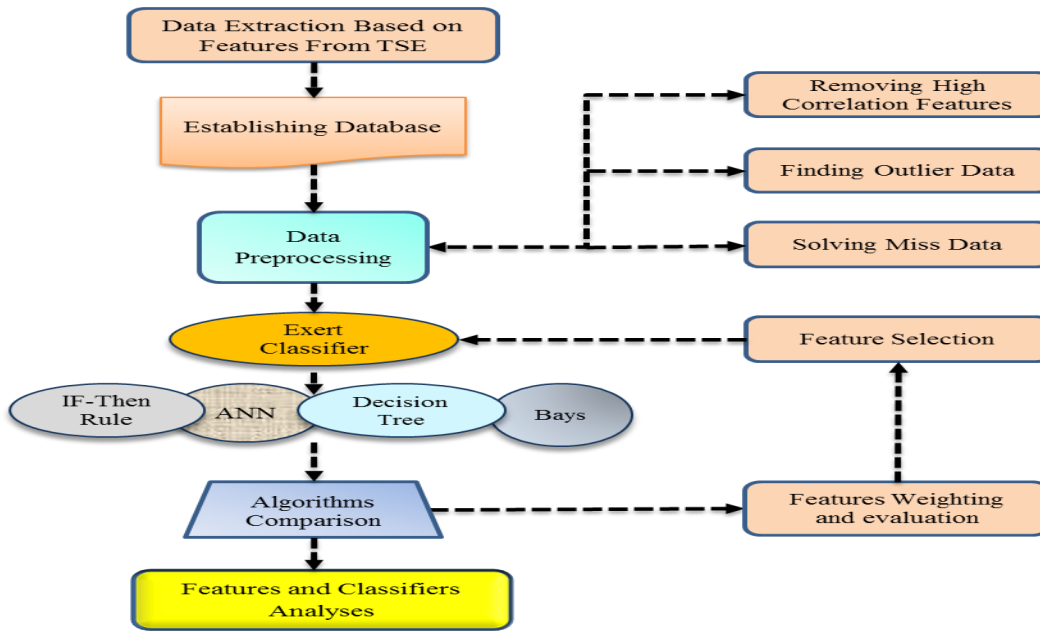


Figure 3.3 Experimental results process

3.3.1. Data pre-processing

- **Removing high correlation features:** features with higher than 95% correlations based on the Pearson test were removed. Therefore, features including gross profit to sale percentage, assess the loan usefulness, cumulative stock profit, fixed assets return percentage, debt to equity ratio, and current debt to equity ratio are removed.
- **Finding outlier data and miss data:** To find outlier data in the database, at first we used the distance-based approach, and by analyzing remote records we concluded that some are very large governmental companies that are not applicable in this chapter and fact they are not outlier data. Other outliers were also deleted. By using the density approach, 12 records were known as Outlier points, in which 7 of them were large companies. Thus they remained in the database. However, others were omitted, mostly because they did not provide accurate information. With clustering approach, we determined some outlier data that were in none of the clusters. As a result, six samples were identified as outlier data. By analyzing the input feature of the company, no suspected case was found, and no company was omitted. Finally, by using the techniques based on the deviation, five records were known as outlier points.

Analyzing records clarified that the input features of these records were pertained to the previous fiscal year and then were removed. Also, among 1963 records, 12 records because of miss value derived from lack of information are deleted.

- **Stationary test:** there are two approaches in stationarity tests; some stationarity tests like the KPSS test, that consider as null hypothesis (H0) that the series is stationary, and the others like the augmented Dickey-Fuller test (ADF) and the Phillips-Perron test (PP), the null hypothesis is on the contrary that the series possesses is not stationary. By implementing KPSS, ADF, and PP tests for return and time series, the p-value of tests which are obtained 0.47, 0.02, 0.011 respectively, clearly the non-stationarity of the time series will be rejected. Thus, the trained models can be reliable for predicting future upcoming horizon

3.3.2. Comparison of algorithms

Table 3-7 shows the results of decision trees, rule-based algorithms, and neural networks accuracy for real return prediction. As it is clear (from Table 3-5), LAD Tree algorithm has achieved higher accuracy in the prediction of return. The other algorithms, like SVM and K-Nearest neighbor, had an accuracy of close to 60%. Thus, due to their low accuracy, we did not apply them for analysis. Low accuracy of SVM algorithm can be because of its high sensitivity to the missed data. It is generally an algorithm to predict 2 class outputs, while we are dealing with multi-class data and many missed data.

Table 3-5 Algorithms Comparison for real return variable

Algorithm	Accuracy	Sensitivity	Specificity	Number of Rules	Tree size	Number of leaves
LAD Tree ⁶ (Mark Hall, et al., 2009)	78.00	77.15	75.29	-	31	15
Cart Decision Tree ⁷	76.50	74.27	74.3	-	13	7
DTNB Rule ⁸ (Mark Hall & Frank, 2008)	76.00	75.08	73.55	998	-	-
Decision Table ⁹	75.50	75.14	72.44	56	-	-
Rep Tree ¹⁰	75.00	74.09	71.64	-	33	17
RIDOR Rules (Witten & Eibe, 2011)	75.00	75.3	73.07	208	-	-
J Rip Rule[36]	74.90	73.18	74.02	9	-	-
BF Tree ¹¹ (Shi, 2007)	74.50	78.49	74.28	-	9	5
Part Rule ¹² (Frank & Witten, 1998)	72.60	67.84	69.15	104	-	-
J 48 Graph	71.50	69.68	66.38	-	1619	810
NB Tree(Witten & Eibe, 2011)	71.00	70.2	69.5	-	97	49

⁶It uses AD Tree with boosting for prediction and cross validation to select training data and class label decisions are done on the basis of this algorithm most votes.

⁷ The used tree's split has been done by Gini index algorithm and the Pruning is done based on cost – complexity after constructing the tree.

⁸At first, these models determined the important variables by using Naive Bays algorithm (18 attribute achieve) and then offer classification prediction rules are provided by decision tree.

⁹ Algorithm first uses the Forward election algorithm to determine the input variables. After 375 implementing the algorithm, ROE, Net working capital, EPS prediction and return are known as effective features and then based on best first (BF) algorithm the model is constructed.

¹⁰This algorithm is based on information gain and its Pruning is based on prediction error minimization.

¹¹ BF Tree uses binary Pruning to construct a tree based on selecting the first important feature as nodes point. Based on this, the model has found the feature that best predicts the output variable among other input variables. Then a binary tree based on this variable is constructed. Best-First (BF) Decision Tree just use return variable to prediction real return. (Just this input is considered in this method's tree).

¹² This method has used C4.5 algorithm in every implantation and has used the best of them as a new rule in the model rules.

LMT Tree(Witten &Eibe, 2011)	70.50			-	30	20
Neural Net (RBF)	70.00	67.3	66.4	-	-	-
Neural Net (MLP ¹³)	69.00	66.3	67.1	-	-	-
Auto MLP ¹⁴	70.00	67.01	66.02	-	-	-
Rule Induction ¹⁵	68.50	69.27	66.76	56	-	-
FT Tree	68.50	66.3	64.8	-	45	28
J 48 Tree	67.39	66.2	65.6	-	303	152
ID3 Numerical	61.50	58.42	57.47	-	1905	979
Bays	60.00	58	62.28	-	-	-
Multinomial Logistic Regression ¹⁶ (Engel, 1988)	42.50	38.72	41.38	-	-	-

From investigating the results, it can be stated that generally denser trees have shown better accuracy than big ones. This is clear in the cases of ID3 Numerical, J48 Graph, and J48 Tree. Due to the importance of pruning after tree construction, since these models have no pruning stage or their pruning algorithm is not efficient, their accuracy is not acceptable.

Trees with a size of less than 33 for real returns (next quarter return) have an accuracy of higher than 70%. Despite the medium accuracy of these trees compared to larger trees, they have higher accuracy in test data. DTNB Rule algorithm has the highest accuracy in comparison with the other “If-Then Rules” algorithms on test data. On average, the accuracy of “If-Then Rules” algorithms are better than trees; however, the best prediction is obtained by LAD Tree. Some algorithms of tree types are shown in Figures 3.4 to 3.7 (Returns in all figures’ nodes are the latest quarter return ration without considering risk).

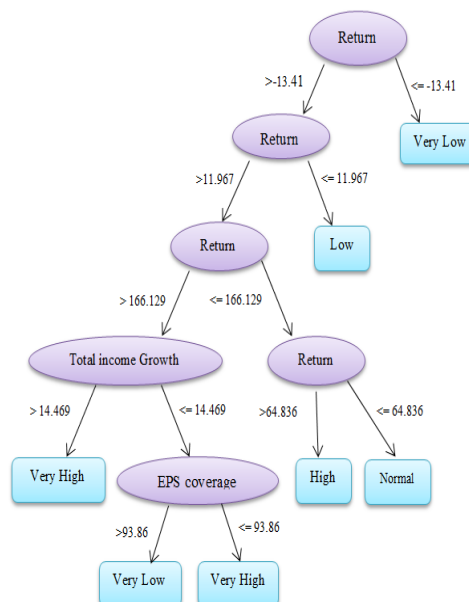


Figure 3.4 Real return prediction - Cart Decision Tree

¹³Feed Forward Multi Layout Perceptron based on Levenberg- Marquat algorithm with 12 Neuron in hidden layer.

¹⁴ Number of neurons and hidden layers are optimized.

¹⁵This algorithm gains the rules based on the information gain and first gained rule is:

If return without risk ≤ 11.967 and return without risk > -12.164 and EPS > 51.447 and EPS coverage percent > 164.500 then Low

This algorithm, gains the rule based on the information gain and based on decrease amount in model accuracy while constructing the rules we prune them. And will construct the model till there is no other variable to be added to the model or the error amount is more than 0.5.

¹⁶ In addition to the ML models, multinomial logistic regression, as a statistical model, is implemented for response variables. Multinomial LR is an extension of LR which can classify multiple dependent variables.

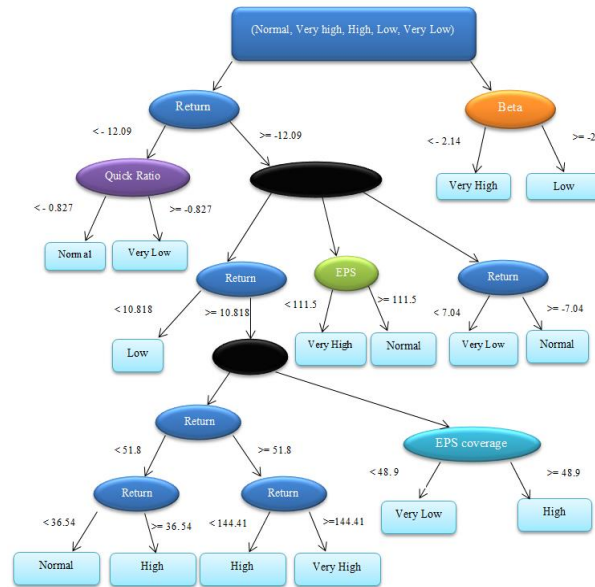


Figure 3.5 Real return prediction - Cart Decision Tree

Figure 3.6. Best-First Decision Tree

0 Return < -13.10281: very low (365.0/62.0)
 0 Return >= -13.10281
 1 Return < 10.81819: Low (392.0/96.0)
 1 Return >= 10.81819
 2 Return < 53.97851: normal (326.0/126.0)
 2 Return >= 53.97851
 3 Return < 136.66997: high (184.0/63.0)
 3 Return >= 136.66997: Very high (72.0/27.0)

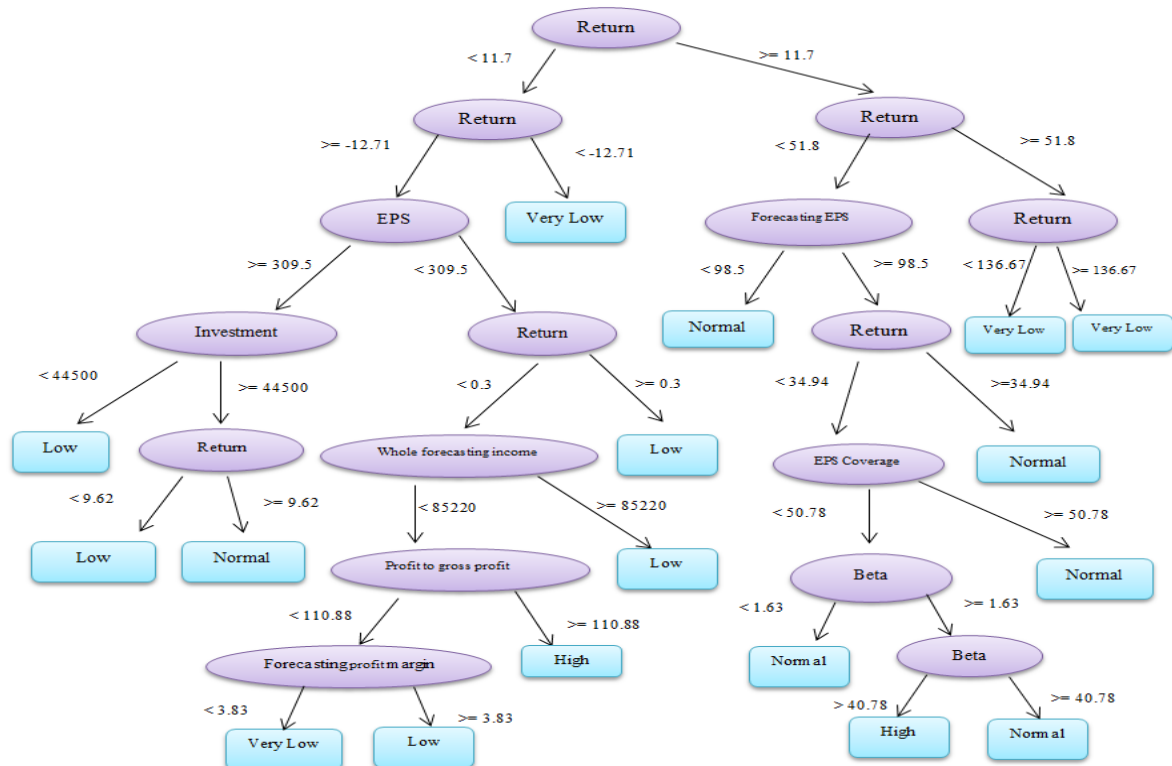


Figure 3.7 Real return prediction - Rep Tree

It is noteworthy to mention that because last quarter performance of the company can be effective for next quarter return (real return) prediction, the return of the stock in the latest quarter is considered as a feature which is shown in Figure 3.4 - 3.7 as Return.

Similarly, the risk is predicted as shown in Table 3-6.

Table 3-6 Algorithm comparison for risk variable

Algorithm ¹⁷	Accuracy	Sensitivity	Specificity	Number of Rules	Tree size	Number of leaves
LAD Tree	78.24	69.62	80.24	-	31	20
DTNB Rule	77.41	69.44	78.51	426	-	-
Decision Table	76.57	65.38	81.32	297	-	-
BF Tree ¹⁸	76.15	67.41	74.71	-	109	55
J Rip Rule	74.90	73.7	74.3	9	-	-
J 48 Graph	73.64	64.68	71.66	-	721	361
Part Rule	73.64	65.36	71.64	55	-	-
Rep tree	72.80	67.13	69.39	-	77	39
Rule Induction	71.55	64.45	70.25	59	-	-
J 48 Tree	71.55	70.9	72.5	-	313	157
FT Tree	67.78	65.6	64.8	-	63	32
NB Tree	66.95	66.3	64.7	-	7	4
Neural Net (MLP)	59.00	61.2	59.22	-	-	-
ID3 Numerical	57.00	55.1	54.2	-	553	403
Bays	55.65	57.3	50.2	-	-	-

It can be stated that generally larger trees (for example higher than 300) and smaller trees have no prominent results in comparison to medium sized trees. In comparison to “If-Then Rules” algorithms, the highest prediction accuracy for test data is gained from DTNB, similar to real return prediction. For risk prediction also “If-Then Rules” algorithms accuracy is also better than those gained by the tree, but the best prediction is gain by LAD Tree. To predict risk, neural network results have lower accuracy in comparison with the prediction of real return. In Figure 3.8 LAD tree for risk prediction is depicted.

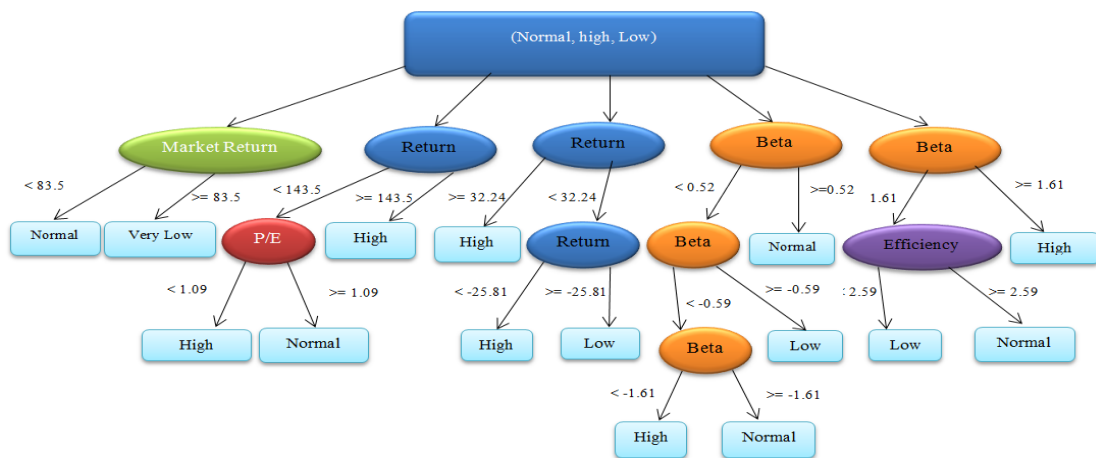


Figure 3.8 Risk prediction- LAD Tree

¹⁷Some algorithms that used in real return prediction have low prediction accuracy in risk prediction and we don't report their results.

¹⁸The beta coefficient is known as the first leaf.

3.3.3. Prediction after hybrid Feature selection

In this section, we develop a hybrid feature selection to evaluate each feature. In the first stage, by using the above-mentioned filter methods, a comprehensive analysis of the features is conducted to assign suitable weights for features. (All features even high correlation features are considered.)

Depending on the evaluation function, we applied the following filter methods:

- Based on the interval method: Relief-f method.
- Based on information: Gini index, Information gain ratio, Information gain, Symmetrical Uncertainty methods.
- Based on correlation: Chi-square, one R methods.
- Based on consistency: Consistency method.
- Based on classified errors: SVM method.

The CFS method did not have any acceptable result because of dependence and was eliminated in the method pre-processing step.

We have applied Rapid Miner and Weka software to implement the algorithms (Hofmann & Klinkenberg, 2013). Furthermore, the algorithm which is used by Weka is mentioned, like Weka IG, Weka Chi-2....

The resulting weights for features of risk parameter and real return parameter are shown in Table 3-7 and Table 3-8, respectively.

Table 3-7 Weighting for risk parameter

Attributes	Chi-2	IG Ratio	Info Gain	R-f	SVM	Con Sistency	Weka IG	Weka chi-2	Weka con	Weka IGR	Weka R-f	Gini Index	Weka oneR
Return	1	0.69	0.61	1	0.024	0.87	0.513	0.437	0.513	0.63	0.9	0.725	0.759
Beta	0.188	0	0.062	0.397	0.079	0.163	0.097	0.073	0.097	0.132	0.519	0.061	0.124
Efficiency	0.091	0.707	0.076	0.071	0.083	0.099	0.056	0.044	0.056	0.112	0.093	0.093	0.127
Market return	0.058	0.67	0.072	0.019	0.023	0.058	0.106	0.092	0.106	0.157	0.021	0.064	0.117
EPS prediction%	0.008	0.615	0.099	0	0.076	0.005	0.05	0.039	0.05	0.17	0.002	0.112	0.111
long-term debt to equity	0.032	0.487	0.076	0.011	0.029	0.039	0.085	0.078	0.085	0.136	0.015	0.077	0.118
Total income growth %	0.021	0.707	0.055	0.005	0.045	0.016	0.03	0.025	0.03	0.16	0.002	0.075	0.095
EPS growth%	0.025	0.707	0.046	0.005	0.061	0.016	0.026	0.021	0.026	0.136	0.002	0.061	0.095
ROE	0.058	0.328	0.074	0.054	0.07	0.073	0.039	0.031	0.039	0.135	0.046	0.092	0.089
DPS	0.046	0.328	0.078	0.047	0.025	0.063	0.041	0.033	0.041	0.15	0.045	0.089	0.123
Debt to total assets ratio	0.07	0.338	0.057	0.035	0.065	0.086	0.03	0.024	0.03	0.141	0.034	0.057	0.1
profit margin growth rate	0.01	0.615	0.036	0.005	0.084	0.008	0.021	0.017	0.021	0.136	0.009	0.045	0.055
EPS	0.046	0.421	0.024	0.063	0.088	0.045	0.016	0.013	0.016	0.056	0.131	0.033	0.093
P/E	0.027	0.328	0.064	0.013	0.108	0.041	0.032	0.025	0.032	0.102	0.002	0.075	0.127
Predicted profit margin	0.053	0.319	0.067	0.071	0.023	0.07	0.036	0.028	0.036	0.11	0.037	0.063	0.047
Stock market value	0.029	0.615	0.023	0.013	0.043	0.029	0.016	0.012	0.016	0.065	0.011	0.027	0.049
ROA	0.033	0.422	0.038	0.01	0.038	0.046	0.023	0.019	0.023	0.152	0	0.056	0.049
current debt to equity ratio	0.039	0.381	0.046	0.003	0.043	0.049	0.026	0.018	0.026	0.121	0.002	0.039	0.092

Attributes	Chi-2	IG Ratio	Info Gain	R-f	SVM	Con Sistency	Weka IG	Weka chi-2	Weka con	Weka IGR	Weka R-f	Gini Index	Weka oneR
debt to equity	0.036	0.419	0.047	0	0.012	0.05	0.041	0.039	0.041	0.114	0.005	0.026	0.043
Book value	0.015	0.615	0.013	0.004	0.131	0.016	0	0	0	0	0.001	0.021	0.048
assess the loan usefulness	0.039	0.421	0.026	0.01	0.031	0.052	0.018	0.016	0.018	0.122	0.006	0.018	0.075
Equity ratio	0.018	0.615	0.003	0.001	0.099	0.019	0	0	0	0	0	0.011	0.04
Gross profit to sale	0.021	0.107	0.055	0.005	0.034	0.016	0.03	0.025	0.03	0.134	0.002	0.074	0.059
Current ratio	0.016	0.615	0.008	0.002	0.073	0.018	0	0	0	0	0	0.013	0.032
Operating profit to sale	0.017	0.615	0	0.001	0.09	0.02	0	0	0	0	0.004	0.002	0.023
P/S	0.014	0.615	0.01	0.002	0.06	0.015	0	0	0	0	0	0.018	0.037
Fixed asset turnover	0.032	0.421	0.006	0.027	0.059	0.024	0	0	0	0	0.128	0.011	0.061
fixed assets return	0.016	0.615	0.008	0.002	0.058	0.018	0	0	0	0	0	0.013	0.032
Debt coverage ratio	0.005	0.366	0.041	0.002	0.045	0.008	0.024	0.019	0.024	0.073	0	0.052	0.088
Liquidity ratio	0.015	0.421	0.028	0.011	0.018	0.022	0.033	0.027	0.033	0.083	0.006	0.025	0
Net profit to sale	0.023	0.377	0.021	0.004	0	0.04	0.015	0.012	0.015	0.046	0.001	0.03	0.068
working capital return percentage	0.002	0.319	0.027	0.012	0.094	0.003	0.017	0.015	0.017	0.064	0.001	0.02	0.049
EPS deviation	0	0.338	0.026	0.006	0.033	0	0.017	0.014	0.017	0.06	0	0.038	0.067
Capital	0.034	0.371	0.016	0.01	0.04	0.036	0	0	0	0	0.025	0.022	0.052
current assets ratio	0.034	0.371	0.016	0.009	0.024	0.037	0	0	0	0	0.025	0.022	0.041
Total predicted income	0.009	0.347	0.024	0.003	0.026	0.015	0.016	0.013	0.016	0.051	0	0.034	0.019
Net profit to gross profit	0.016	0.338	0.009	0.03	0.068	0.019	0	0	0	0	0.027	0.015	0.013
EPS coverage percent	0.008	0.328	0.016	0.014	0.061	0.016	0	0	0	0	0.003	0.024	0.063
Net working capital	0.015	0.358	0.009	0.019	0.011	0.025	0	0	0	0	0.001	0.013	0.064
Average payment	0.018	0.319	0.011	0.034	0.024	0.034	0	0	0	0	0.013	0.007	0.053
Total asset turnover	0.015	0.358	0.008	0.019	0.011	0.025	0	0	0	0	0.001	0.013	0.054
Quick ratio	0.027	0.328	0.009	0.001	0.043	0.038	0	0	0	0	0.01	0.015	0.027
Stock cumulative profit	0.012	0.338	0.004	0.005	0.016	0.018	0	0	0	0	0.001	0.008	0.033
Current assets turnover	0.012	0	0	0.01	0.073	0.019	0	0	0	0	0.007	0	0.051

Table 3-8 Weighting for real return parameter

Attributes	Chi-2	IGR	IG	R-f	SVM	Con consistency	Weka IG	Weka chi-2	Weka con	Weka IGR	Weka R-f	Gini Index	Weka oneR
Return	1	0.69	0.61	1	0.024	0.87	0.513	0.437	0.513	0.63	0.9	0.725	0.759
Market return	0.188	0	0.062	0.397	0.079	0.163	0.097	0.073	0.097	0.132	0.519	0.061	0.124
ROA	0.091	0.707	0.076	0.071	0.083	0.099	0.056	0.044	0.056	0.112	0.093	0.093	0.127
Beta	0.058	0.67	0.072	0.019	0.023	0.058	0.106	0.092	0.106	0.157	0.021	0.064	0.117
ROE	0.008	0.615	0.099	0	0.076	0.005	0.05	0.039	0.05	0.17	0.002	0.112	0.111
EPS growth %	0.032	0.487	0.076	0.011	0.029	0.039	0.085	0.078	0.085	0.136	0.015	0.077	0.118
Profit margin growth rate	0.021	0.707	0.055	0.005	0.045	0.016	0.03	0.025	0.03	0.16	0.002	0.075	0.095
Operating profit to sale	0.025	0.707	0.046	0.005	0.061	0.016	0.026	0.021	0.026	0.136	0.002	0.061	0.095
EPS	0.058	0.328	0.074	0.054	0.07	0.073	0.039	0.031	0.039	0.135	0.046	0.092	0.089
DPS	0.046	0.328	0.078	0.047	0.025	0.063	0.041	0.033	0.041	0.15	0.045	0.089	0.123
EPS prediction%	0.07	0.338	0.057	0.035	0.065	0.086	0.03	0.024	0.03	0.141	0.034	0.057	0.1
Predicted profit margin	0.01	0.615	0.036	0.005	0.084	0.008	0.021	0.017	0.021	0.136	0.009	0.045	0.055
Net profit to sale	0.046	0.421	0.024	0.063	0.088	0.045	0.016	0.013	0.016	0.056	0.131	0.033	0.093
EPS deviation	0.027	0.328	0.064	0.013	0.108	0.041	0.032	0.025	0.032	0.102	0.002	0.075	0.127
Efficiency	0.053	0.319	0.067	0.071	0.023	0.07	0.036	0.028	0.036	0.11	0.037	0.063	0.047
P/S	0.029	0.615	0.023	0.013	0.043	0.029	0.016	0.012	0.016	0.065	0.011	0.027	0.049
Net profit to Gross profit	0.033	0.422	0.038	0.01	0.038	0.046	0.023	0.019	0.023	0.152	0	0.056	0.049

Attributes	Chi-2	IGR	IG	R-f	SVM	Con sistency	Weka IG	Weka chi-2	Weka con	Weka IGR	Weka R-f	Gini Index	Weka oneR
Quick ratio	0.039	0.381	0.046	0.003	0.043	0.049	0.026	0.018	0.026	0.121	0.002	0.039	0.092
Equity ratio	0.036	0.419	0.047	0	0.012	0.05	0.041	0.039	0.041	0.114	0.005	0.026	0.043
stock market value	0.015	0.615	0.013	0.004	0.131	0.016	0	0	0	0	0.001	0.021	0.048
Book value	0.039	0.421	0.026	0.01	0.031	0.052	0.018	0.016	0.018	0.122	0.006	0.018	0.075
Long-term debt to equity	0.018	0.615	0.003	0.001	0.099	0.019	0	0	0	0	0	0.011	0.04
Gross profit to sale	0.021	0.107	0.055	0.005	0.034	0.016	0.03	0.025	0.03	0.134	0.002	0.074	0.059
Debt to equity ratio	0.016	0.615	0.008	0.002	0.073	0.018	0	0	0	0	0	0.013	0.032
Debt coverage ratio	0.017	0.615	0	0.001	0.09	0.02	0	0	0	0	0.004	0.002	0.023
Current debt to equity	0.014	0.615	0.01	0.002	0.06	0.015	0	0	0	0	0	0.018	0.037
Net working capital	0.032	0.421	0.006	0.027	0.059	0.024	0	0	0	0	0.128	0.011	0.061
Assess the loan usefulness	0.016	0.615	0.008	0.002	0.058	0.018	0	0	0	0	0	0.013	0.032
Stock cumulative profit	0.005	0.366	0.041	0.002	0.045	0.008	0.024	0.019	0.024	0.073	0	0.052	0.088
P/E	0.015	0.421	0.028	0.011	0.018	0.022	0.033	0.027	0.033	0.083	0.006	0.025	0
Liquidity ratio	0	0.615	0.007	0.007	0.025	0	0	0	0	0	0.001	0.012	0.02
working capital return percentage	0.023	0.377	0.021	0.004	0	0.04	0.015	0.012	0.015	0.046	0.001	0.03	0.068
Total income growth%	0.002	0.319	0.027	0.012	0.094	0.003	0.017	0.015	0.017	0.064	0.001	0.02	0.049
current ratio	0	0.338	0.026	0.006	0.033	0	0.017	0.014	0.017	0.06	0	0.038	0.067
Current assets ratio	0.034	0.371	0.016	0.01	0.04	0.036	0	0	0	0	0.025	0.022	0.052
Debt to total assets ratio	0.034	0.371	0.016	0.009	0.024	0.037	0	0	0	0	0.025	0.022	0.041
Current assets turn over	0.009	0.347	0.024	0.003	0.026	0.015	0.016	0.013	0.016	0.051	0	0.034	0.019
Total asset turn over	0.016	0.338	0.009	0.03	0.068	0.019	0	0	0	0	0.027	0.015	0.013
EPS coverage percent	0.008	0.328	0.016	0.014	0.061	0.016	0	0	0	0	0.003	0.024	0.063
fixed assets turn over	0.015	0.358	0.009	0.019	0.011	0.025	0	0	0	0	0.001	0.013	0.064
Total predicted income	0.018	0.319	0.011	0.034	0.024	0.034	0	0	0	0	0.013	0.007	0.053
Fixed assets return	0.015	0.358	0.008	0.019	0.011	0.025	0	0	0	0	0.001	0.013	0.054
Average payment	0.012	0.338	0.004	0.005	0.016	0.018	0	0	0	0	0.001	0.008	0.033
capital	0.012	0	0	0.01	0.073	0.019	0	0	0	0	0.007	0	0.051

After attaining the features weights by different filter based algorithms we have 13 columns (m) with 44 attributes' weight (n) and then because of obtaining accurate clusters, we use preprocessing on this data set. Then, we use the clustering-function-based method for clustering attributes to predict the most important features. Usually, the first and second clusters are the effective ones and we choose them(Li, 2006b).

For real return, in the first step of clustering, (last quarter) return and the market return are separated from the other attributes. In other words, they are more important with higher weights compared with other attributes. In the second step, 6 features out of 43 remaining ones are separated. Finally, eight features in the first and second clusters are considered as important features. Similarly, for risk parameter in the first step, three features, return, beta coefficient and efficiency of 44 features can be separated. In the second step, 12 features out of the 42 remaining ones can be separated. Finally, 15 features from 44 features are selected as important ones. The results for risk and real return parameters are presented in Table 3-9.

Table 3-9 Selected features for risk and real return parameters

Selected features of the first and second cluster, based on function clustering method for real return parameter.	Return, Market return, Beta coefficient, Return on asset (ROA), Percent of growth EPS, EPS, Predicted profit margin, EPS coverage percent.
Selected features of the first and second cluster, based on function clustering method for risk parameter.	Return, Beta coefficient, Efficiency, Market return, EPS prediction, Percent of growth EPS, DPS, P/E, EPS, Equity ratio, Stock book value, Debt to total assets ratio, Predicted profit margin, P/S, Total incomes Growth

As can be seen, more features were selected for the risk variable than with real return. The classification results with selected features show in parenthesis at Table 3-10, in which “Deviation = Accuracy base on selected feature - Accuracy base on all feature”. If the deviation is positive, it means that use of important feature will improve the prediction results and vice versa.

Table 3-10 Algorithms Deviation

Algorithm	Risk Accuracy Deviation	Return Accuracy Deviation
LAD Tree	2% (80.24%)	1% (79%)
Cart Decision Tree	1.5% (66.5%)	-4.5% (72%)
DTNB Rule	-0.9% (76.51%)	-1% (75%)
Decision Table	-1.2% (75.37%)	0.07% (76.20%)
BF Tree	-2% (74.15%)	1.5% (76%)
J Rip Rule	0% (74.90%)	-1.2% (73.7%)
J 48 Graph	-1.83% (71.81%)	2% (69.50%)
Part Rule	1.91% (75.55%)	2% (74.6%)
Rep tree	0.77 % (73.52%)	-2% (73%)
Rule Induction	0.95% (72.50%)	1. 5% (70 %)
J 48 Tree	-1.5% (70. 05%)	-0.50% (66.89%)
FT Tree	2% (69.18%)	2 % (70.5%)
NB Tree	-4.20% (62.75%)	-1% (70%)
Neural Net (MLP)	2% (61.00%)	2.5% (71.5%)
ID3 Numerical	2.5% (59.5%)	2% (63.5%)
Bays	-1.40% (54.15%)	-2 % (58.00%)

As results show from Table 3-10, by this hybrid method we can get better prediction in some methods with fewer numbers of features.

3.4.Discussion

The real return results in prediction with selected features:

If for denser structure trees all effective features in the first prediction are selected by the proposed hybrid model, results in better accuracy, such as “BF tree”, “LAD tree”, and “FT tree”. Otherwise, it is possible that accuracy drops, like “CART and Rep” trees. The selected features have a different effect on the accuracy of forecasting. Some trees with a large structure, such as J48 graph and J48 Tree are got lower accuracy, while some get a higher accuracy such as ID3 Numerical. Higher accuracy of all algorithms is due to the fact that the hybrid feature selection model, as a pruning algorithm, is used to reduce over training error. Bays algorithms for both real return and risk obtained weak prediction. Thus, DTNB and also NB Tree output for both real return and risk achieved lower accuracy. The results show that the rule base algorithms with average number of rules, such as Part Rule, Decision Table and Rule Induction, obtain better results.

On the other hand, the accuracy of the algorithms with fewer rules like J Rip Rule has descended. The accuracy of the neural network for each output has increased, because of not getting stuck in local optimum points.

The risk results in prediction with selected features:

Due to the large number of features extracted from the hybrid feature selection algorithm for risk, the moderate size tree, such as Rep tree, FT Tree, and LAD tree have better accuracy than before. However, BF Tree accuracy has been decreased because of removing 2 effective attribute.

With this analysis, it is also clear that the algorithms, such as LAD Tree, which use the features beta coefficient, market return, P/E, and the efficiencies, obtained the highest accuracy which has improved up to 80.24%. Large trees such as J 48 Tree and J 48 Graph get lower prediction accuracy but ID3 Numerical results are improved. As said before, the prediction result of bays based algorithms like DTNB and NB Tree have been decreased but the large drop in NB Tree prediction is because of its dense structure.

For other rule-based algorithms, the prediction result has been improved or remained stable, except the decision table algorithm. This is derived from the average number of rules that are covered by the selected features.

Moreover, by using the weight of features obtained from Chi-2 or IG Ratio or Info Gain algorithms (without using the hybrid model), the return and market returns features to predict real return get the highest weight (90% of cumulated weight). Maybe, the high percentages predicted by BF Tree and LAD Tree algorithms are due to these two features. Also to risk parameter, return, beta coefficient, and efficiency features get the highest weight (90% of cumulated weight). Thus the high accuracy of LAD Tree, FT Tree, Rep Tree, and Rule Induction algorithms could be due to these three features. Because these algorithms emphasize these high weight features more than others.

A comparison between our method and similar researches is illustrated in Table 3-11. Six hybrid methods which have a brilliant accuracy in return forecasting in different country stock exchange compared based on input data, base classifier, feature selection, hybrid prediction model and the best accuracy as follow:

Table 3-11 Comparison results with other studies

Author /Year	Stock Exchange	Input Data	Base Classifier	Feature selection	Hybrid Model	The Best Accuracy%
Tsai, et al. (2011)	Electronic Industry in Taiwan	19 Financial ratios and 11 Macroeconomic indicators	MLP- Cart - Logistic Regression	---	Bagging - Voting	66.67
C.-F. Huang (2012)	30 special companies in Taiwan	14 Financial ratios	SVR- GA	---	---	85- 76.71
Cheng, et al. (2010)	Taiwan	10 Technical Indexes and 8 Macroeconomic indicators	PNN- C4.5- Rough Set	---	Hybrid	76
Huang, et al. (2008)	South-Korea and Taiwan	23 Technical Indexes	SVM- K-NN- Cart- Logistic Regression- Back Propagation	Wrapper	Voting	76.06 80.28
Tsai and Hsiao (2010)	Taiwan	8 Fundamental Index and 11 Macroeconomic indicators	---	GA-PCA-Cart	Back Propagation	79
(Tsai, Lu, and Yen (2012))	Taiwan	61 intangible assets value variable	MLP	PCA- Stepwise Regression- decision trees- association rules- GA	MLP	75
Recent work	Return Forecasting in TSE-Iran	44 Financial ratios and Fundamental Index	Cart , Rep Tree, LAD Tree, ...	Function based Clustering	Hybrid	80.24
Recent work	Risk Forecasting in TSE-Iran	44 Financial ratios and Fundamental Index	DTNB , BF Tree, LAD Tree,...	Function based clustering	Hybrid	79.01

We also exerted data dimension reduction methods including Principle Component Analyses (PCA), Independent Component Analysis (ICA), Factor Analysis (FA), Discrete Wavelet Transform (DWT), and Discrete Fourier Transform (DFT) methods on data set. Our results on these methods show that despite of long runtime the accuracy of prediction algorithms highly decreased. As an instance, after the reduction of dimensionality from 44 to 11 with PCA algorithm, the LAD Tree prediction results get 52.7 % which is very low accuracy.

Moreover, the data reduction process time in this data is very high, and as an instance, based on Rapid miner Software it takes 11 hours and 32 minutes in DWT algorithms. Although by using the MATLAB algorithm, the execution time is less than before, but the accuracy of the results will not differ much. Among these 5 algorithms, ICA results despite of long execution time (approximately 31 hours with Rapid miner software) obtain better prediction accuracy and the accuracy predicted based on LAD Tree is 71%.

3.5.Conclusions

In this chapter, an approach for prediction of risk and real return was developed by applying data mining techniques as well as fundamental data set. To do this, first through a comprehensive study, the features which can be potentially effective on risk and return were investigated. Then, after developing an appropriate database, the preprocessing of the database step was taken. To predict the real return and risk, 20 and 15

different prediction algorithms were applied respectively. Then, the strength and weakness of each one were investigated by analyzing the size and leaves of tree algorithms or/and “If-Then Rules” gains of rule-based algorithms. In the next step, by using a hybrid feature selection algorithm based on 9 different filter algorithms and function-based clustering method, important features were selected and re-prediction with selected features was performed. The results show that for real return parameter, the number of effective features is usually less than the number of effective features on the risk parameter. With the help of these features, the results in most algorithms were improved. In this way, this hybrid feature selection method is capable of identifying effective features. The high accuracy of prediction results indicates that the extracted features explain the behavior of the market very well and can be considered as a suitable database for future research. These findings can enable the investors to analyze the market and gain high accurate results with fewer features, and not getting confused in the market by many features which are not necessarily effective. The novelty of this approach is consideration of the combination of 9 different feature selection algorithms with function-based clustering algorithm. This hybrid model can enjoy the advantages of all feature selection algorithms and make a robust and accurate decision. The effectiveness of the model is illustrated with the prediction of both risk and return of stocks and then analyzing the results with and without implementing the hybrid feature selection algorithms. While almost none of the relevant studies in this field pay attention to the prediction of risk feature. Furthermore, a systematic and efficient methodology is designed for comprehensive searching the potential representative features on the stock market in 3 categories of financial ratio, profit & loss reports, and Stock pricing models and not arbitrary choosing likely effective features. The limitation of this method is that collecting all data and information may be difficult for some real cases. Future research directions include but are not limited to:

1. Combining prediction methods in the framework of fusion models or optimize the classification algorithms by applying some meta-heuristics algorithms to improve the prediction results
2. Predicting the other important variable (in addition to risk and return) such as liquidity (Barak, et al., 2013).
3. Using technical features and textual information, in addition to fundamentals features, in order to have more comprehensive features and to be able to predict short term situation of stocks
4. Customizing the proposed approach for the prediction of risk and return in a particular industry or investigating the accuracy of the procedure by data from other popular stock markets, such as US stock market which may result in new dimensions in this procedure.
5. Applying different clustering models to our feature selection data set and compare results by considering new feature selection methods, such as CFS (X. Zhang, Hu, Xie, Wang, et al., 2014) or entropy-based clustering for feature selection (H.-Y. Lin, 2013).

Chapter 4

Fusion of Multiple Diverse Predictors in Stock Market

Summary of the fourth chapter

As mentioned in chapter three, forecasting stock returns and their risk represents one of the most important concerns of market decision makers. Although many studies have examined single classifiers of stock returns and risk methods, fusion methods, which have only recently emerged, require further study in this area. The main aim of this chapter is to propose a fusion model based on the use of multiple diverse base classifiers that operate on a common input and a Meta classifier that learns from base classifiers' outputs to obtain a more precise stock return and risk predictions. A set of diversity methods, including Bagging, Boosting, and AdaBoost, is applied to create diversity in classifier combinations. Moreover, the number and procedure for selecting base classifiers for fusion schemes are determined using a methodology based on dataset clustering and candidate classifiers' accuracy. The results demonstrate that Bagging exhibited superior performance within the fusion scheme and could achieve a maximum of 83.6% accuracy with Decision Tree, LAD Tree and Rep Tree for return prediction and 88.2% accuracy with BF Tree, DTNB and LAD Tree in risk prediction. For feature selection part, a wrapper-GA algorithm is developed and compared with the fusion model. Chapter four seeks to help the researcher select the best individual classifiers and fuse the proper scheme in stock market prediction. To illustrate the approach, we apply it to the Tehran Stock Exchange (TSE) data for the period from 2002 to 2012.

4.1.Introduction

In this chapter similar to chapter three, the prediction of stock return and risk are implied separately based on fundamental features in order to build a comprehensive model for stock market analysis. Although the statistical approaches such as logistic regression and regression analysis are widely applied to forecast the return and risk of stocks, the results of machine learning approaches are generally superior in comparison to statistical methods (Barak & Modarres, 2015; S.-H. Cheng, 2014, 2015). The multiple classifier ensemble systems (MCS), one type of machine learning technique, has recently become the focus of a new methodology for obtaining higher accuracy in predictions. The rationale is that the optimization of a combination of relatively simpler predictors appears more convenient than optimizing the design of a single complex predictor (Barak & Sadegh, 2016; Haghighi, et al., 2011). Three fundamental issues are effective for establishing a successful MCS model: accuracy of individual classifiers, diversity among classifiers, and

the choice of the fusion methods that will be used. The aim of this combination scheme is to gain increased precision with proper single classifiers and eliminate the uncorrelated individual classifier errors, which are the errors made by individual classifiers on various parts of input space (Tsai, et al., 2014).

In fusion methods, multiple dissimilar predictors are used and combined by a fusion algorithm that combines the outputs of the individual predictors. Fusion methods in MCS are generally categorized as linear, non-linear, statistical, and machine learning combination methods. Linear methods are known as the simplest fusion methods. For instance, the sum and average of the individual classifiers' outputs are examples of linear fusion methods (L. I. Kuncheva, 2004). Non-linear methods include rank-based combiners such as Borda Count and majority voting strategies (L. I. Kuncheva, 2004; Sharky & Sharky, 1997). In statistics-based fusion methods, statistical techniques such as regression or Bayesian combination methods are used to combine the outputs of individual classifiers (Pereira, Mitchell, & Botvinick, 2009). Finally, different machine learning methods such as decision trees (DT) and support vector machines (SVM) can be used to fuse the base learner.

The second fundamental issue, diversity, refers to the differences existing among decisions made by various classifiers. In classifier combination design, it is believed that the success of combinations not only depends on the individual classifiers' suitability but also on diversity being inherent among them. In fact, classifiers that are strong in different areas are supposed to be diverse. The entire point of fusing multiple classifiers is to balance the weaknesses of the individual classifiers. This balancing requires classifiers that make errors in different areas of the decision space. Diversity creation methods are generally categorized as explicit and implicit methods (Haghighi, et al., 2011). Explicit methods generally seek to optimize certain metrics during diversity creation. Boosting and AdaBoost (Yoav Freund & Schapire, 1996) are examples of explicit diversity methods that directly manipulate the training data distributions in order to make some sort of diversity in the combination procedure. Implicit methods, unlike explicit methods, pay no special attention to diversity metrics. Bagging, as an implicit method, randomly samples from training data in order to train each classifier individually and these samples will be reused to produce diverse combination members (Breiman, 1996).

In this chapter, in which we aim to improve the accuracy of the risk and return prediction of stocks, we propose a fusion model framework that relies on the combination of multiple dissimilar and diverse classifiers operating on a common input. In the first phase, cross-validation is applied on the dataset, and a specific (but optimum) number of different classifiers sets is learned from the dataset (creating a pool of classifiers). A classifier selection procedure is proposed in which the dataset is first clustered by the k -means method, after which the optimum number of clusters is chosen by Streamlined Silhouette Criterion Average (SSCA). The performance of the classifiers on the dataset is then evaluated, and the best combinations are

selected for the fusion phase. Finally, in the fusion phase, Bagging, Boosting and AdaBoost are applied to the classifiers of the selected combinations of the previous phase and one fusion algorithm as a Meta-classifier learns from their predictions to provide the final prediction of the initial input data.

The contribution of the chapter is summarized as follows:

- Designing a fusion model for returns and risk prediction of stocks in the financial market.
- Applying various diversity methods in order to achieve more precise predictions.
- Considering the simultaneous risk and return prediction of stocks.
- Developing a base classifier selection procedure from candidate procedures by dataset clustering and considering the accuracy of combined classifiers.
- Developing a wrapper-GA scheme for feature selection and prediction and comparing it with the fusion method.

This chapter is divided into six sections and organized as follows: The backgrounds of multi-classifier systems and combination models, as well as diversity creation and fusion methods, are discussed in section 4.2. In Section 4.3, the proposed multi-classifier ensemble systems (MCS) is generally discussed. Next, in section 4.4, the experimental results are provided, and the discussion of real return and risk prediction with the proposed fusion model is presented in section 4.5. Moreover, in this section, a new selection scheme is also developed for the feature selection part and compared with the hybrid method. Finally, this chapter's conclusions and future research directions are presented in section 4.6.

4.2.Literature Review

4.2.1. Stock prediction with classifier ensembles

The best way to design of MCS to achieve higher accuracy has become an important research topic in the field of pattern recognition, as stated in several related review articles (Finlay, 2011; Oza & Tumer, 2008; Rokach, 2009). The main idea behind using ensembles is that the combination of classifiers can improve the performance of a pattern recognition system in terms of better generalization with increased efficiency and clearer design (D. F. de Oliveira, Canuto, & de Souto, 2009). Wolpert (2001) believes that every classifier has its specific competencies over other competing algorithms, and MCS tries to take advantage of each of the available trained classifiers based on their competencies for different parts of the feature space.

Dasarathy and Sheela (1979) proposed combining a linear classifier and a k -nearest neighbour classifier in which the conflicting feature space regions were first identified by classifiers, after which one classifier works on the features of the conflicting region and the other works on the remaining features. This can be considered as the first study suggesting a classifier selection concept for MCS design. In 1981, Rastrigin and Erenstein (1981) further developed the idea by partitioning the feature space into several regions and

assigning the individual classifiers with the best accuracy over each region. A survey of multiple classifier systems as hybrid systems can be found in (Woźniak, Graña, & Corchado, 2014).

In the stock exchange and financial research area, different machine learning methods such as artificial neural networks (ANN), decision trees (DT) and support vector machines (SVM) are widely applied to establish efficient ensemble systems. Neural network ensemble systems are found to be effective in achieving superior accuracy for stock price forecasting (Lahmiri, 2014; Tsai, et al., 2011; L. Wang & Wu, 2012). In addition to neural networks, algorithms based on decision trees use a greedy search approach and tend to choose a search direction using a heuristic attribute evaluation function (Salzberg, 1994). This approach, however, does not guarantee to find an optimal solution. Thus, a combined algorithm starting from different initial points in the search space can improve the DT's performance in finding an optimal model. Qian and Rasheed (2007) combined several machine learning classifiers, including artificial neural networks, decision trees, and k -nearest neighbour, to design a hybrid model for stock market prediction. The results show that the accuracy of up to 65% is achieved. Tsai, et al. (2014) combined three classifiers, including multi-layer perceptron (MLP) neural networks, SVM and DT based on a combination of the bagging and boosting methods. Their results showed that DT ensembles utilizing boosting techniques have the best performance, and this superiority is demonstrated by further studies on a Taiwan bankruptcy dataset.

4.2.2. Fusion in ensembles scheme

Fusion methods refer to the approaches used to obtain classifier ensembles. The motivation behind such methods is to combine predictions so that misclassification is less likely to occur. In other words, a proper fusion method is the one that can exploit the strength of individual classifiers and optimally combine their outputs to provide the final decision of the system (Woźniak, et al., 2014). Among early studies regarding fusion techniques for MCS, the majority voting schemes application is widely known (L. Kuncheva, Whitaker, Shipp, & Duin, 2003).

In another category, aggregation methods such as supremum, average, mean or median value perform simple fusion operators and lack any learning procedures (Fumera & Roli, 2005; Kourentzes, Barrow, & Crone, 2014; Wozniak, 2008). The essential advantage of these methods is that they balance the over-fitting of the individual classifiers. Tumer and Ghosh (1996) used a large number of unbiased and independent classifiers and reported the average of the outputs as the final results. These researchers claimed that their method returned the same results as applying the optimal Bayes classifier. Ho, Hull, and Srihari (1994) used different methods based on decision ranks, such as Borda counts, for the combination function in order to achieve a useful representation for each classifier's decision. A Borda count is categorized as a support function in a fusion system that assigns a score for the decisions taken from each classifier.

Machine learning fusion methods are another group of fusers that use the accuracy of individual classifiers as training data and then apply a learner algorithm, e.g., DT, K-NN,..., as a high level classifier (meta-classifier) that learns from the accuracy of individual classifiers to obtain higher accuracy (Ferreiro, Sierra, Irigoien, & Gorritxategi, 2011).

4.2.3. Diversity and accuracy creation

System diversity is highlighted as a crucially important aspect of MCS design (LudmilaI Kuncheva & Whitaker, 2003; Tang, Suganthan, & Yao, 2006; Tsymbal, Pechenizkiy, & Cunningham, 2005). The main purpose of designing MCS is to integrate a set of mutually complementary individual classifiers in order to achieve outputs with higher accuracy and diversity and less correlation. Diversity can generally be achieved by inducing variations in classifier parameters (e.g., weights and topology of a neural network as initial parameters) (Windeatt, 2005), classifiers' training datasets (e.g., using the learning strategies, such as Bagging and Boosting) (L. Kuncheva, 2004), and classifier types (e.g., using different types of classifiers as ensemble members).

Variation in classifiers' training dataset (data partitioning) is important for several reasons, such as data privacy or learning procedure requirements over distributed data partitions in different databases. In this category, cross-validation is a well-known approach that minimizes overlap among dataset partitions (Krogh & Vedelsby, 1995). Bagging (Breiman, 1996) and Boosting (Y. Freund, 1995; Schapire, 2001) are both known as the most popular techniques in diversity creation and originate in bootstrapping. In this chapter, diversity is achieved by making variations in the training dataset using the abovementioned techniques (e.g., cross-validation and Bagging, Boosting) and using different classifier types (e.g., neural network, decision trees, and SVM) as ensemble members.

4.3. Proposed Multi-Classifer Ensemble System (MCS)

The MCS is generally composed of three main phases: (1) generation, (2) selection, and (3) integration (see (Alceu S. Britto, Sabourin, & Oliveira, 2014)). The generation phase focuses on creating a pool of base classifiers composed of the most appropriate candidates for the subsequent classifier selection and integration steps. In the second phase, the best classifiers from the pool are selected for building the MCS, and finally, in the integration phase, the predictions of the selected classifiers are combined in order to make a final decision.

In the generation phase, the aim is to create as many diverse classifiers as possible. Thus, several diversity methods are applied to build a pool of diverse base classifiers. In the second and third phases, a meta-learning approach is proposed (Giraud-Carrier, 2008). In other words, the selection of the most competent classifiers and final classifications are considered as another classification problem, termed the meta-problem. For a

given instance, the base classifiers' outputs and the real class of the sample are passed down to a Meta classifier as its input data, and then the Meta classifier estimates the final class of the given sample.

In the following, three phases of the proposed MCS framework, including the generation of the base classifiers (pool of classifiers), selection of competent classifiers and final fusion phase are widely discussed.

4.3.1. Phase one: Base classifier generation

The main aim of MCS design is to select competent classifiers and combine their predictions effectively. In order to ease and increase the accuracy of the selection procedure, it is preferable to build an initial set of potentially competent base classifiers (pool of classifiers) first and then decide whether each candidate is sufficiently qualified to classify an input instance. A key factor here is to combine the prediction results of those classifiers in which the decision boundaries are widely different. To achieve this, the creation of diversity among classifiers is proposed. A diversification strategy aims to train the classifiers on different (disjoint) input subspaces in order to create a set of different but complementary classifiers. Several diversification approaches that are mainly used for generating diverse classifiers are discussed by (Mousavi & Eftekhari, 2015). Using unstable classifiers such as decision trees (DT) and neural networks (NN), applying different sets of classifiers including NN, DT, SVM, etc., and utilizing various datasets for training the base classifiers are the diversification methods used in this chapter. To create variations in classifiers' training dataset, four methods are applied: Cross-validation, Bagging, Boosting and AdaBoost, each of which is discussed in detail below.

- ***Cross-validation***

Similar to Chapter 3, in order to evaluate the robustness of the predictor and minimizes the overlapping of dataset partitions, a 10-fold cross-validation is applied, and the best result is selected.

- ***Bagging***

Bootstrap aggregation, simply known as Bagging, is an ensemble-based algorithm and one of the most intuitive and simple to implement methods with extremely good performance. In the Bagging approach, the classifier is trained on different training datasets that are generated by the bootstrap method (Breiman, 1996). The Bootstrap method builds k training datasets by randomly re-sampling the original given dataset with replacement. Thus, there are k independent training datasets for the classifier training procedure. When classifier training is complete, the final results should be aggregated via an appropriate method, such as majority voting. Pseudo code for the Bagging method is given as follows:

Algorithm 4.1 : Bagging**Input:**

Training data \mathcal{S} with correct labels $\omega_i \in \Omega = \{\omega_1, \dots, \omega_C\}$ representing C classes

Weak learning algorithm **WeakLearn**

Integer T specifying the number of iterations

Percent (of fraction) F to create bootstrapped training data

Do: $t=1, \dots, T$

1. Take a bootstrapped replica \mathcal{S}_t by randomly drawing percent of \mathcal{S} .
2. Call WeakLearn with \mathcal{S}_t and receive the hypothesis (classifier) h_t .
3. Add h_t to the ensemble, E .

End**Test: Simple Majority Voting** - Given unlabeled instance x

1. Evaluate the ensemble on x .
2. Let $v_{t,j} = \begin{cases} 1, & \text{if } h_t \text{ picks class } \omega_j \\ 0, & \text{otherwise} \end{cases}$
3. Obtain total vote received by each class $V_j = \sum_{t=1}^T v_{t,j}$, $j = 1, \dots, C$
4. Choose the class that receives the highest total vote as the final classification.

- **Boosting**

In Boosting method, it is believed that finding many prediction rules can be much easier than building one rule with a high level of accuracy. In boosting, unlike bagging, each classifier is trained on different k training sets in a sequential and not a parallel and independent way. The algorithm creates an ensemble of classifiers through data resampling in order to provide the most informative training data for each consecutive classifier. Boosting creates three weak classifiers: the first classifier, C_1 , is trained by a random sample of the training data. For the second classifier, C_2 , the training set is selected as the most informative subset. In other words, half of the training data for C_2 is correctly classified by C_1 and the other half is misclassified by C_1 . Finally, the third classifier C_3 is trained on samples that are misclassified by both C_1 and C_2 . In fact, in each iteration, the classifier creates a new set of prediction rules and assigns new weights to data so that the classifier will pay more attention to misclassified tuples in subsequent iterations. Finally, after many repetitions, the boosting algorithm combines these rules into one single prediction rule that is expected to be much more accurate than any of the single rules.

Ultimately, boosting places heavier weights on the samples that are most often misclassified in every round. In other words, it forces the base learner (i.e., the individual classifier) to focus on the hardest samples that were mostly misclassified by the preceding rules. In order to combine the achieved prediction rules, using a (weighted) majority voting approach is suggested as an efficient method (Schapire, 2001). The pseudo code for the Boosting algorithm is given as follows:

Algorithm 4.2: Boosting

Input:

Training data \mathcal{S} of the size N with correct labels $\omega_i \in \Omega = \{\omega_1, \dots, \omega_C\}$;

Weak learning algorithm **WeakLearn**.

Training

1. Select $N_I < N$ patterns without replacement from \mathcal{S} to create data subset \mathcal{S}_I .
2. Call **WeakLearn** and train with \mathcal{S}_I to create classifier C_I .
3. Create dataset as the most informative dataset \mathcal{S}_2 , given C_I , such that half of \mathcal{S}_2 is correctly classified by C_I , and the other half is misclassified. To do so:
 - a. Flip a fair coin. If Heads, select samples from \mathcal{S} , and present them to C_I until the first instance is misclassified. Add this instance to \mathcal{S}_2 .
 - b. If Tails, select samples from \mathcal{S} and present them to C_I until the first one is correctly classified. Add this instance to \mathcal{S}_2 .
 - c. Continue flipping coins until no more patterns can be added to \mathcal{S}_2 .
4. Train the second classifier C_2 with \mathcal{S}_2 .
5. Create \mathcal{S}_3 by selecting those instances for which C_I and C_2 disagree. Train the third classifier C_3 with \mathcal{S}_3 .

Test – Given a test instance x

1. Classify x by C_I and C_2 . If they agree on the class, this class is the final classification.
2. If they disagree, choose the class predicted by C_3 as the final classification

• **AdaBoost**

In AdaBoost (Yoav Freund & Schapire, 1997), the most popular boosting algorithm, a series of models are combined; in each model, the dataset is re-sampled and weighted based on their difficulty to be learned and

classified (J. Cao, Kwong, & Wang, 2012). AdaBoost takes the training set $S_n = [(x_1, y_1), \dots, (x_m, y_m)]$ as input data and calls the base learning algorithm repeatedly for a series of iterations $t = 1, \dots, T$. The $w_t = \{w_t^1, w_t^2, \dots, w_t^n\}$ represents the weight distribution over samples in iteration t and is equally distributed in the first iteration. In each iteration t , AdaBoost maintains the weights on the training sample i denoted as w_t^i so that the weights of misclassified samples will increase and the learner algorithm will pay more attention to these difficult samples in the training set in subsequent iterations. Based on the given w_t in each round t , the base learning algorithm finds the most appropriate classifier h_t and assigns an importance measure α_t to it, set as:

$$\alpha_t = \frac{1}{2} \ln\left(\frac{1-e_t}{e_t}\right) \quad \text{Eq. 4-1}$$

where e_t represents the mean squared error (MSE) for h_t . The final classifier H will be built by a weighted majority vote of the T base classifiers in which the parameter α_t is assigned as the weight of classifier h_t . The pseudo code of the presented AdaBoost is given as follows (Barak & Sadegh, 2016):

Algorithm 4.3: AdaBoost

Input: Initial training set composed of n samples, denoted as $S_n = [(x_1, y_1), \dots, (x_m, y_m)]$

Initialize: $w_1^i = 1/n$, i.e., $w_1 = \{w_1^1, w_1^2, \dots, w_1^n\} = \{1/n, 1/n, \dots, 1/n\}$

For $t = 1, 2, \dots, T$

Take R_t samples randomly from S_n

Determine the weight distribution w_t

Build a classifier h_t using R_t as the training set

Compute: $e_t = \text{MSE for } h_t$ and $\alpha_t = \frac{1}{2} \ln\left(\frac{1-e_t}{e_t}\right)$

Update the distribution weight set: $w_{t+1}^i = \text{normalize}(w_t^i * \exp(-\alpha_t))$

Output: The combined classifier: $H = \sum_{t=1}^T \alpha_t h_t$

4.3.2. Phase two: Classifier selection

In addition to diversity, the accuracy of the base classifiers is another critical factor in base classifier selection (Woloszynski & Kurzynski, 2011; Woloszynski, Kurzynski, Podsiadlo, & Stachowiak, 2012) and can guarantee the effectiveness of the MCS.

The proposed selection phase seeks to discover sets of classifiers (fusion set) that improve the classification accuracy in integration. In order to specify the optimum number of classifiers in fusion sets, it is suggested to determine the optimum number of clusters of the dataset first and then find the sets of classifiers with the same number. To achieve this, the k-means clustering algorithm is first applied to the dataset for different values of k , after which the best value is chosen.

The k-means algorithm is one of the most straightforward unsupervised learning methodologies, using a simple method to classify a given dataset into a certain number of clusters (k clusters). The aim is to determine k centroids that should be cunningly placed as different locations resulting in different outputs. Thus, it is better to place these centroids as far as possible from each other. Next, each point of the dataset should be associated with the nearest centroid. When no point is left, the early grouping is complete. At this point, k new centroids must be recalculated for the center of clusters built in the previous step. A new binding must be carried out on the dataset points toward the new k centroids. This loop continues until no more changes occur in centroids' locations and they do not move any further. For a given set of observations ($\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$), where each observation is a d -dimensional real vector, the best condition in k-means clustering is to partition the n observations into k ($\leq n$) sets $\mathbf{S} = \{\mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_k\}$ in which the within-cluster sum of squares is minimized:

$$\arg \min_S \sum_{i=1}^k \sum_{x \in S_i} \|x - \mu_i\|^2 \quad \text{Eq. 4-2}$$

where μ_i is the mean of points in S_i .

The k-means algorithm is then performed for different values of k . In order to specify the optimum number of clusters (k), the streamlined silhouette criterion average (SSCA) is calculated (Covões & Hruschka, 2011). SSCA evaluates the closeness of each object of a cluster to the objects of nearby clusters. For the i th observation of the k th cluster, the Silhouette index is defined as follows:

$$s_{i,k} = \frac{b_{i,k} - a_{i,k}}{\text{Max}(b_{i,k}, a_{i,k})} \quad \text{Eq. 4-3}$$

where $b_{i,k}$ is the minimum average distance between observation i and all other observations to the nearest neighbouring clusters except the k -th cluster, and $a_{i,k}$ represents the average distance between observation i and the remaining $(n_k - 1)$ observations of the k -th cluster. The most effective clustering is the one that results in maximum SSCA, computed as:

$$SSCA = \frac{1}{k} \sum_{j=2}^k \frac{1}{n_j} \sum_{i=1}^{n_j} s_{i,j} \quad \text{Eq. 4-4}$$

The optimum number of clusters (k) resulting from SSCA calculations determines the number of base classifiers in an ensemble that creates the widest decision boundaries. The proposed classifier selection process is illustrated in Figure 4.1. The cross-validation is first performed on training data for diversity creation among base classifiers. In order to increase the base classifiers' accuracy, the heuristic Meta cost method is applied. This method intensifies the error of prediction by multiplying it by an integer number. In other words, if the classifier predicts class A for an input data but the real class is two steps above or below the predicted class, Meta cost penalizes the error by doubling it. Similarly, if the real class is three steps away from the predicted one, the error will be multiplied by three. Table 4-1 illustrates the error penalty scores of the Meta cost method.

Based on the optimum value of k achieved by SSCA, all possible sets of k classifiers are formed and their outputs, including the real class of the samples, are passed down as inputs to the meta-classifier. For each set of classifiers, the accuracy index is calculated. If it is higher than 75%, the related set is selected; otherwise, it will be removed from the set of classifiers.

The accuracy of the meta-classifier is defined as the percentage of the correctly predicted sets of tuples on the given dataset. For a five-class prediction problem, the accuracy can be measured using a confusion matrix, as shown in Table 4-2 with the associated formula:

$$Accuracy = \frac{a_1 + b_2 + c_3 + d_4 + e_5}{\sum_{i=1}^5 a_i + b_i + c_i + d_i + e_i} \quad \text{Eq. 4-3}$$

Table 4-1 Meta cost penalty score matrix

		Real class				
		1	2	3	4	5
Predicted class	1	0	1	2	2	3
	2	1	0	1	2	2
	3	2	1	0	1	2
	4	2	2	1	0	1
	5	3	2	2	1	0

Table 4-2 Confusion matrix for a five-class prediction problem

		Predicted class				
		Very low	Low	Normal	High	Very high
Actual class	Very low	a_1	b_1	c_1	d_1	e_1
	Low	a_2	b_2	c_2	d_2	e_2
	Normal	a_3	b_3	c_3	d_3	e_3
	High	a_4	b_4	c_4	d_4	e_4
	Very high	a_5	b_5	c_5	d_5	e_5

The proposed method considers the accuracy and diversity of the candidate classifiers simultaneously and results in a set of competent classifiers for the subsequent integration stage. The entire selection phase is formalized in Algorithm 4.4 as follows:

Algorithm 4.4: Classifier selection phase**Input:**

Pool of classifiers, denoted as $P = \{c_1, c_2, \dots, c_M\}$;

Training dataset, denoted as *Train_data*;

Optimum number of clusters achieved by k -means and SSCA, denoted as k ($k \leq M$).

Output: Set of selected base classifiers denoted as Bc .

Algorithm.

```
1:   $Bc = \emptyset$ 
2:  Perform 10-fold cross-validation on Train_data.
3:  for all  $c_i \in P$  do
4:    Train  $c_i$  by Train_data.
5:    Perform Meta cost algorithm.
6:  end for
7:  for  $j = 1:C_k^M$ 
8:    Form  $Set_j$  with  $k$  members taken from  $P$  without replacement.
9:    Achieve the outputs of classifiers in  $Set_j$  and the real class of sample data, and pass them down to the Meta-classifier.
10:   if Average Accuracy > 75%
11:      $Bc = Bc \cup Set_j$ 
12:   end if
13: end for
14: Return  $Bc$ 
```

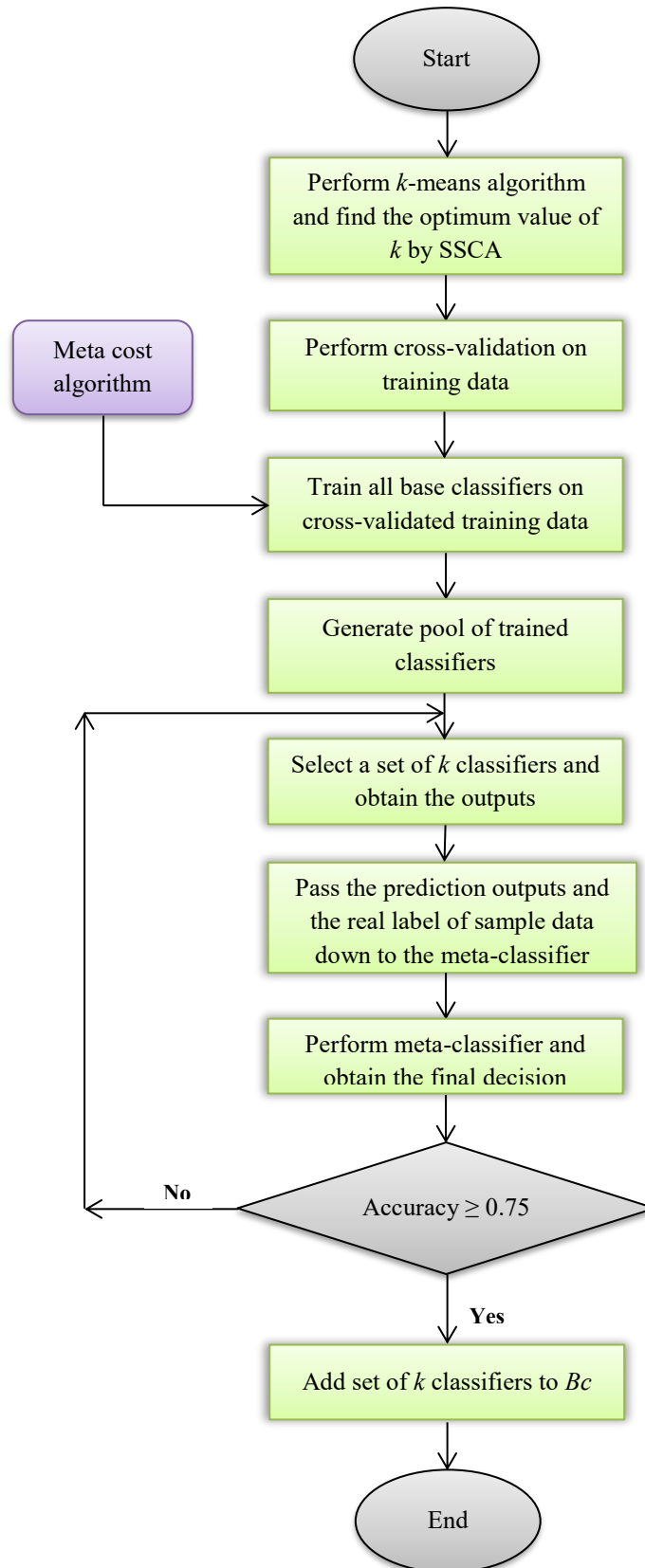


Figure 4.1 Selection phase process

4.3.3. Phase three: Fusion

In the generation phase, different diversity algorithms were introduced for the initial generation and training of classifiers. The cross-validation is used in the selection phase and the remaining algorithms, including Bagging, Boosting, and AdaBoost, will be used for this final fusion phase. Because different machine learning algorithms have dissimilar errors, the fusion of multiple different classifiers is expected to decrease the overall error rate.

- ***Fusion scheme: training and operation***

Apart from the initial diversifications, the fusion scheme is similar to the previous selection phase. In order to train the fusion scheme, the three above mentioned diversity algorithms are first used to train the classifiers of every selected set in Bc . Then, for a given sample dataset, the classification results, including the real class, are passed down as inputs to the meta-classifier. Based on the achievements of base classifiers and the real response, the meta-classifier will return the final classification of the given sample data.

Suppose we have a set of k stock classifiers, C_k , with $1 \leq k \leq K$. The feature vector, X_i , for the i th instance ($1 \leq i \leq I$) that is used for training all C_k of set j ($1 \leq j \leq J$) is defined as:

$$X_i = [\alpha_1, \alpha_2, \dots, \alpha_p, \dots, \alpha_P], i = 1, 2, \dots, I \quad \text{Eq. 4-4}$$

where α_p is the p -th feature ($1 \leq p \leq P$) of the feature vector X_i .

When all the C_k are trained, they are fed with the test data, and the outcomes are the set of individual classifiers' predictions for i -th instance, y_i^k , which are defined as:

$$y_i^k = f_{C_k}(X_i), i = 1, 2, \dots, I \quad \text{Eq. 4-5}$$

The vector Y_i is formed by individual predictions, y_i^k , with the real label of the i -th instance, R_i , being used to train the upper-level meta-classifier.

$$Y_i = \{y_i^k\}^T, i = 1, 2, \dots, I \quad \text{Eq. 4-6}$$

Once the meta-classifier is trained, the fusion scheme is prepared for further operations.

For each set j of Bc , the feature vector of the i -th instance of test data, X_i , is given to the k individual classifiers, C_k , as input data. The outputs, y_i^k , form the prediction vector, Y_i , and together with the real label, R_i , are used as input data for the meta-classifier. The final decision for i -th instance is made as given by:

$$D_i = g(Y_i, R_i), i = 1, 2, \dots, I \quad \text{Eq. 4-7}$$

for some real values of D_i . Figure 4.2 shows the generalized flowchart of the fusion phase.

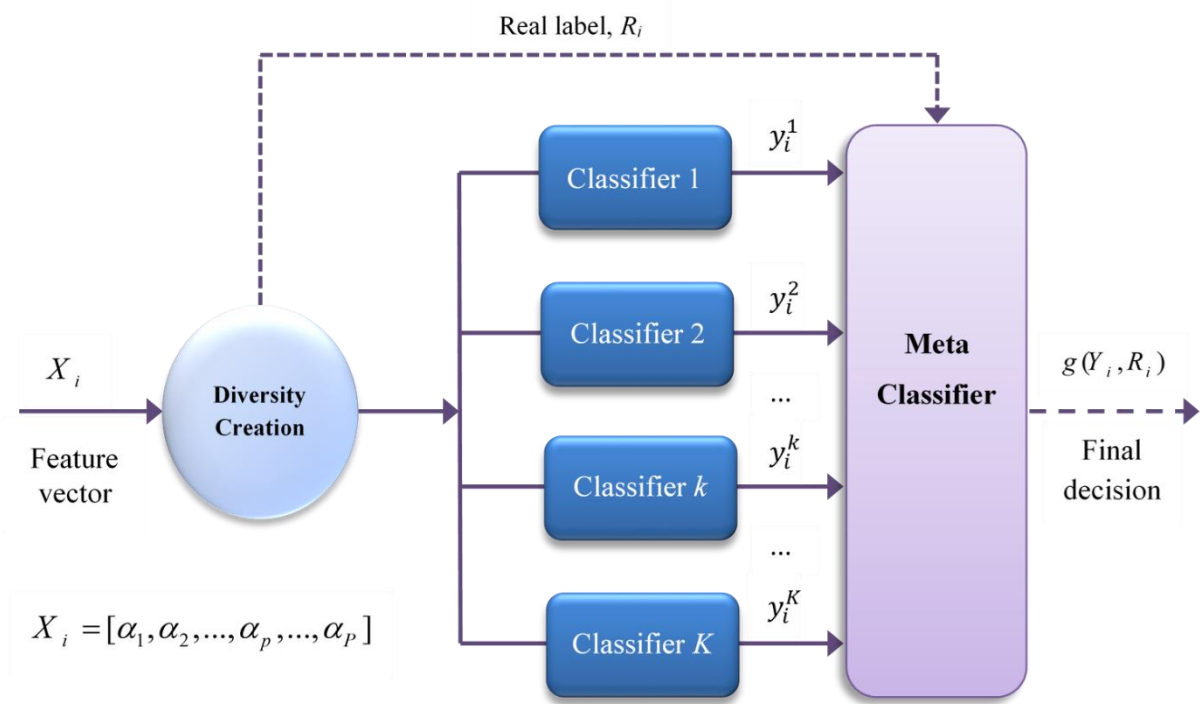


Figure 4.2 Generalized flowchart of proposed Fusion phase

4.4. Experimental Results

In this chapter, the same dataset of Chapter 3 is used for implementing fusion model which includes data from the Tehran Stock Exchange (TSE) from 2002 to 2012 for a total of 1963 records for 400 companies. Additionally, the same specification for real returns and risk data that are created in Chapter 3 are applied including five intervals for real return (very high, high, average, low, and very low) and three range for risk. The pre-processing section of Chapter 3, including filtering and clustering approaches for detecting outlier data are applied, and outlier data records are omitted. Besides, to use the best features for forecasting section, the selected features of the hybrid approach of Chapter 3 which are gathered in Table.3.3 are implemented to forecast the real return (next period return) and risk as response variables. The next sections will provide detail of the proposed experiment, which is shown in Figure 4.3.

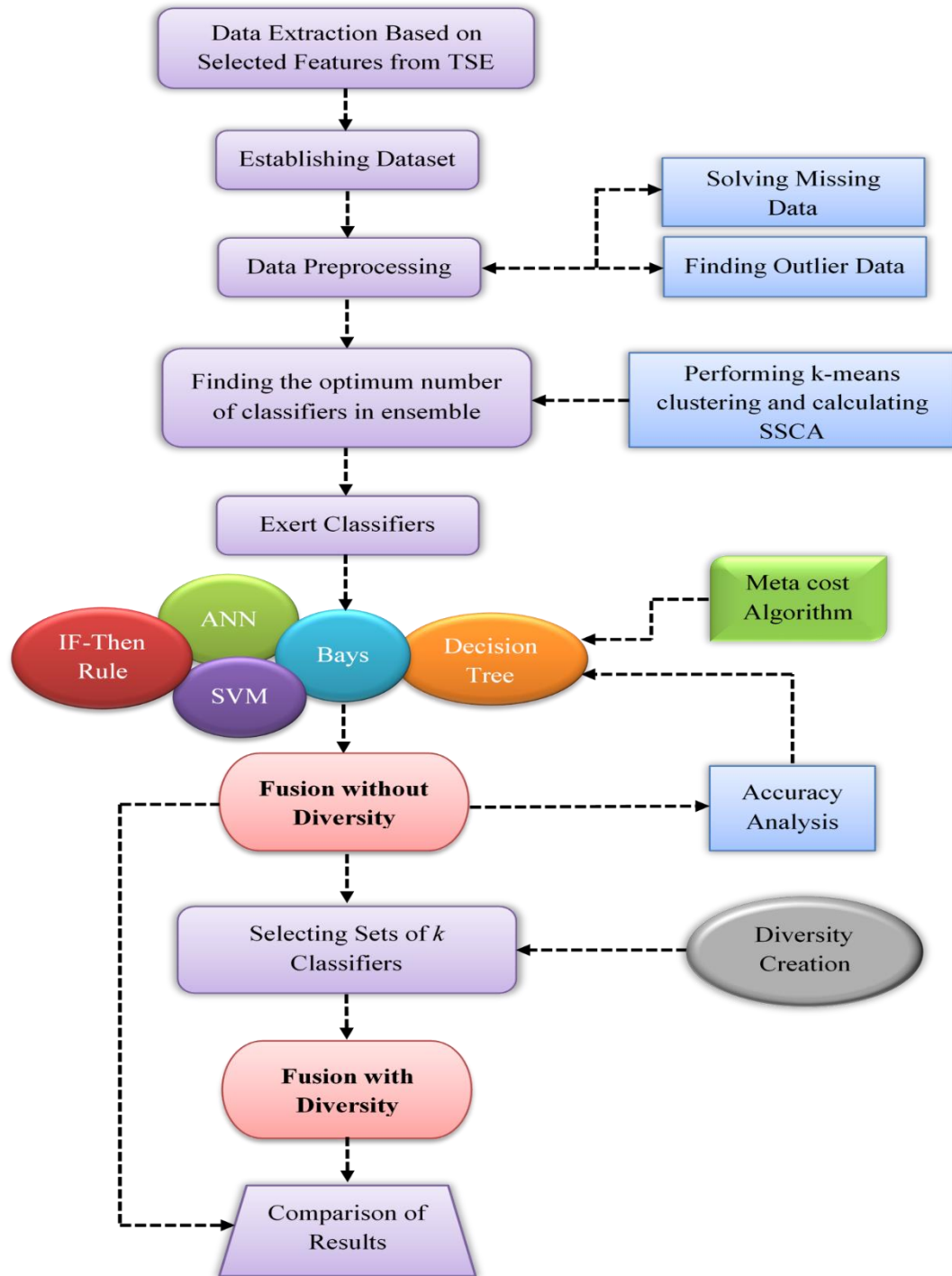


Figure 4.3 The proposed process of stock return and risk

4.4.1. Individual classifiers

In stock prediction, the complexity of the data necessitates applying models that are capable of defining such intricacy. Different methods, including statistical methods and neural networks, are studied, and it is found that the results gained by machine learning and data mining algorithms are much more prominent

(Patel, Shah, Thakkar, & Kotecha, 2015a). In this chapter, Decision Trees (DT), Artificial Neural Networks (ANN), rule-based algorithms and Support Vector Machines (SVM) are used as the individual classifiers.

Decision trees and rule-based algorithms are known to be powerful prediction algorithms with outstanding performance in stock return prediction (P. Ou & H. Wang, 2009). Rule-based classifiers use a set of IF-THEN rules for classification, which is especially useful when there are specific relationships among input variables. A decision tree is composed of decision rules that separate the independent variables into homogeneous areas and build rules that can be used for the output prediction of a set of input variables. The rules obtained from this group of classifiers are of importance to investors seeking to make their best portfolio. In this study, the LAD tree, Cart decision tree, Rep tree, BF tree and certain other popular, rule-based algorithms such as decision tree naive Bayes (DTNB) rule, J RIP rule, RIDOR rule, and Part Rule are used.

Artificial Neural Networks (ANNs) are a group of analytical techniques capable of approximating extremely sophisticated non-linear functions. The multi-layer perceptron (MLP) is a popular neural network architecture that is a strong function estimator for prediction or classification problems. The MLP can learn complex non-linear functions to an arbitrary level of accuracy.

Support Vector Machines (SVMs) are the third group of classifiers used in this research. SVMs are categorized as generalized linear models that achieve the classification/regression decision using features linear combination. The approximated function in SVMs can be either a classification function, which is also used for the data categorization of this research or a regression function for estimating the numerical value of input data.

4.4.2. Selection of classifier sets

In the second phase, sets of classifiers with size k should be generated. In order to specify the optimum number of classifiers (k), the k -means algorithm is first performed with different values of k . Then, the SSCA is calculated for each cluster based on Euclidean distance. The results in Table 4-3 shows that if we increase the number of clusters, the maximum amount of SSCA will be achieved in $k=3$, after which its value starts to decrease. Thus, using $k=3$ and based on Algorithm 4.3, sets of classifiers with three members are generated that are then used in the final fusion phase.

Table 4-3 SSCA calculations on different number of clusters

Number of clusters	2	3	4	5
SSCA	12.27	14.43	11.38	9.74

The performance of each classifier for forecasting the return and risk of stocks on the given dataset are presented in Chapter 3. In order to specify the meta-classifiers, the algorithms that provide greater prediction

accuracy on the given dataset are chosen as meta-classifiers. Investigations state that trees and rule-based algorithms with a denser structure generally display better accuracy than larger ones. Thus, the meta-classifiers are chosen from such dense and accurate algorithms.

The 10-fold cross-validation is performed, and the average accuracy of classifier sets with 3 members for stock return and risk predictions are given in Table 4-4 and Table 4-5, respectively. These selected sets are specified as the best ensembles whose performance with three diversity methods is analyzed in the fusion step. The results reveal that although some of the algorithms did not individually display an accurate performance (e.g., SVM, Bayes, and MLP), they could achieve an acceptable level of accuracy in an ensemble model.

Table 4-4 Average accuracy percentage of selected sets for stock return prediction (without diversity).

Set Number	Base classifiers			Fusion algorithm	Accuracy
1	Decision Tree	LAD Tree	BF Tree	DTNB	78.69%
2	BF Tree	Bays	SVM	LAD Tree	78.04%
3	Decision Tree	LAD Tree	Rep Tree	BF Tree	77.34%
4	Decision Table	DTNB	Rep Tree	Decision Tree	76.23%
5	MLP	Part Rule	J48 Graf	BF Tree	75.84%

Table 4-5 Average accuracy percentage of selected sets for stock risk prediction (without diversity).

Set Number	Base classifiers			Fusion algorithm	Accuracy
1	Decision Tree	LAD Tree	DTNB	Decision Table	78.94%
2	DTNB	LAD Tree	BF Tree	Decision Table	78.34%
3	Part Rule	Decision Tree	MLP	BF Tree	77.10%
4	Decision Tree	LAD Tree	BF Tree	DTNB	77.01%
5	Rep Tree	LAD Tree	BF Tree	Decision Table	76.90%
6	Rep Tree	Decision Table	DTNB	LAD Tree	75.43%
7	Bays	SVM	Part Rule	LAD Tree	75.16%

4.4.3. Stock return and risk predictions with the proposed fusion scheme

In the final phase, as illustrated in Figure 4.2, three diversity algorithms (Bagging, Boosting and AdaBoost) are performed separately for each selected set, as shown in Table 4-4 and Table 4-5, and the confusion matrix of the highest accuracy for risk and return predictions are reported in Table 4.7 and Table

4.9, respectively. To make more reliable results with more baseline algorithms, Random Forest as well as K-Nearest Neighbours (K-NN), SVM, and Bayes algorithms are also implemented.

For the real return prediction, the highest prediction accuracy is estimated as 83.65%, which is achieved by the Decision Tree, LAD Tree and Rep Tree ensemble with BF Tree as its fusion algorithm and Bagging as its diversity method. The detailed prediction results of the superior ensemble are presented in Table 4-7.

Table 4-6 Average accuracy percentage of stock return prediction with diversity creation.

Base classifiers			Diversity	Fusion algorithm	Accuracy
Decision Tree	LAD Tree	BF Tree	Bagging	DTNB	80.05%
Decision Tree	LAD Tree	BF Tree	Boosting	DTNB	66.55%
Decision Tree	LAD Tree	BF Tree	AdaBoost	DTNB	64.50%
Decision Tree	LAD Tree	Rep Tree	Bagging	BF Tree	83.65%
Decision Tree	LAD Tree	Rep Tree	Boosting	BF Tree	70.02%
Decision Tree	LAD Tree	Rep Tree	AdaBoost	BF Tree	68.12%
Decision Table	DTNB	Rep Tree	Bagging	Decision Tree	82.23%
Decision Table	DTNB	Rep Tree	Boosting	Decision Tree	79.32%
Decision Table	DTNB	Rep Tree	AdaBoost	Decision Tree	77.72%
MLP	Part Rule	J48 Graph	Bagging	BF Tree	78.74
BF Tree	Bays	SVM	Bagging	LAD Tree	70.66
K-NN	SVM	Bayes	Bagging	Decision Tree	64.27%
Random Forest			-	-	73.05%

Table 4-7 Prediction results of the best ensemble for real return prediction.

Accuracy: 83.65%	True Very low	True High	True Very high	True Normal	True Low	Class precision
pred. Very low	14	1	0	1	3	73.16%
pred. High	0	48	2	3	0	90.57%
pred. Very high	1	2	19	2	1	76.00%
pred. Normal	0	9	1	56	1	83.67%
pred. Low	2	1	1	1	31	86.11%
Class recall	82.00%	78.69%	82.60%	88.88%	86.11%	

As shown in Table 4-9, the highest accuracy of 88.23% in risk prediction is achieved by the BF Tree, DTNB and LAD Tree ensemble with Decision Table as the fusion algorithm and Bagging as the diversity method. The detailed prediction results of this ensemble are presented in Table 4-9.

Table 4-8 Average accuracy percentage of stock risk prediction with diversity creation.

Base classifiers			Diversity	Fusion algorithm	Accuracy
BF Tree	DTNB	LAD Tree	Bagging	Decision Table	88.23%
BF Tree	DTNB	LAD Tree	Boosting	Decision Table	78.90%
BF Tree	DTNB	LAD Tree	AdaBoost	Decision Table	80.27%
BF Tree	Decision Tree	LAD Tree	Bagging	DTNB	81.34%
BF Tree	Decision Tree	LAD Tree	Boosting	DTNB	78.56%
BF Tree	Decision Tree	LAD Tree	AdaBoost	DTNB	80.12%
BF Tree	Rep Tree	LAD Tree	Bagging	Decision Table	82.46%
BF Tree	Rep Tree	LAD Tree	Boosting	Decision Table	66.51%
BF Tree	Rep Tree	LAD Tree	AdaBoost	Decision Table	70.55%
DTNB	Decision Tree	LAD Tree	Bagging	Decision Table	79.14%
DTNB	Decision Table	Rep Tree	Bagging	LAD Tree	76.43%
Part Rule	Bays	SVM	Bagging	LAD Tree	68.66%
MLP	Decision Table	Part Rule	Bagging	BF Tree	80.56%
K-NN	SVM	Bayes	Bagging	Decision Tree	69.71%
Random Forest			-	-	74.36%

Table 4-9 Prediction results of the best ensemble for risk prediction.

Accuracy: 88.23%	True High	True Low	True Normal	Class precision
pred. High	246	3	8	95.72%
pred. Low	14	388	32	89.40%
pred. Normal	65	54	685	85.20%
Class recall	75.69%	87.19%	94.48%	

Based on the results in Table 4-6 and Table 4-8, bagging generally displayed better performance in improving the prediction accuracy of ensembles in comparison with Boosting and AdaBoost; however, it could not effectively improve the prediction accuracy of ensembles with weak base classifiers. In order to predict the real return and risk of stocks, voting methods such as simple averaging and weighted averaging are also used as the meta-classifier algorithms, which resulted in maximum accuracy of 80% for real return prediction, although performance was deficient for risk prediction.

4.5. Discussion

A comparison of the findings presented in Table 4-6 and Table 4-8 show the significant role of applying Bagging as a diversity method on accuracy improvement in the real return prediction. Bagging enhanced the accuracy of almost all ensembles except the ensemble of BF Tree, Bayes, and SVM with LAD Tree

fusion algorithm. The reason may be the weakness of Bayes and SVM algorithms (in the dataset) used as base classifiers, whose performance could not be improved even with Bagging.

Moreover, based on findings shown in Table 4-6 and Table 4-8, the performance of almost all stock classifiers has greatly improved in fusion system with the exception of the LAD Tree and Decision Tree, which provided higher accuracy only in concert with the Bagging diversity algorithm.

The results achievements show that the fusion of the strongest individual classifiers has led to more accurate prediction, but this is not necessarily always true. For instance, the accuracy of MLP is 69% (see Chapter 3 for individual forecasters' results), but in fusion with Part Rule and J48 Graph algorithms, the accuracy improved to 78.74%. One reason for this may be that the weakness of some algorithms can be covered by other classifiers used in the fusion system. In other words, the algorithms can demonstrate complementary behaviour when they are combined in this way.

Among several classification algorithms used in this chapter, the SVM and Bayes algorithms demonstrated different behaviour. Although their accuracies in both real return and risk prediction were improved by the fusion system, the findings revealed that the fusion system without diversity resulted in better accuracy. The high sensitivity of SVM to the missed data individually resulted in low accuracy of 60% for real return prediction, but the ensemble with the Bayes and BF Tree algorithms diversified by Bagging led to an accuracy of 70.66%; without diversity creation, an accuracy of 78.04% was achieved.

Similar to the real return, the achievements on the risk prediction also revealed that the fusion system with diversity successfully improved the performance of individual classifiers in all combinations except for the SVM and Bayes algorithms, whose performance was not improved by diversity creation. The superiority of Bagging in improving accuracy in comparison with the other two diversity algorithms shows its high consistency with the proposed prediction procedure.

Because of the low accuracy, the fusion results of some sets of classifiers with Boosting and AdaBoost methods are not reported in Table 4-6 and Table 4-8.

As stated above, Bagging method significantly outperformed Boosting and AdaBoost methods. An important question to ask is why Bagging outperforms other methods and what exactly differentiates it from Boosting and AdaBoost. One reason for its superior performance may be the level of "sensitivity" displayed by each of these algorithms towards the degree of clearance in the dataset. The existence of missing values and outlier data is inevitable in problems with large datasets. Although the pre-processing functions decrease the effects of such deficiencies, they are not capable of completely resolving it. The findings reveal that Boosting and AdaBoost are more sensitive to the robustness of the dataset, whereas Bagging is scarcely affected by such imperfections.

Another reason for the significant outperformance of Bagging may be related to the instability of the prediction procedure. As Breiman (1996) notes, Bagging is highly capable of improving the accuracy of unstable prediction procedures where a small change in training data can result in large changes in the predictor/classifier structure. Neural networks, classification, and regression trees are specified as unstable methods in which Bagging works well. The achievements also show the superiority of Bagging to the other two algorithms in similar circumstances.

In this chapter, in addition to the hybrid method of Chapter 3, a Wrapper-GA algorithm is developed for the feature selection part. In this algorithm, a population is generated from candidate solutions, and in each iteration, a new set of individuals is generated using mutation and crossover functions. The fitness of individuals in the current population is evaluated by a Decision Tree algorithm, and the best individuals are selected as the next generation. The overview of the proposed Wrapper framework is presented in Figure 5.1 (refer to the next chapter).

In this methodology, after the pre-processing stage, the GA selects a set of features and then tests the prediction error of the selected features using the CART decision tree algorithm. Ten-fold cross-validation is applied in order to generate the training and test data. GA uses a uniform mutation with a probability of 0.1, tournament algorithm for selecting children (crossover) with a rate of 0.8, and a population size of 60. The Decision Table is set as the fitness function. Additionally, the GA was run with varying parameter values (population size, mutation and crossover rates, etc.) in order to enhance the accuracy. Because the Decision Table algorithm is sensitive to missing data, these missed values are substituted with the average value of that column.

Table 4-10 gives the selected features with the best accuracy by Wrapper-GA for real return prediction.

Table 4-10 Selected features for real return prediction with Wrapper-GA algorithm.

Average payment period	current assets turnover	predicted profit margin	Equity ratio
quick ratio	EPS prediction	profit margin growth rate	β coefficient
return on equity (ROE)	Debt ratio	EPS growth percentage	percentage of net profit to the sale
prediction difference percentage of EPS with the real amount	Return on asset (after tax) ROA	long-term debt to equity ratio	P/E, EPS cover

The accuracy of real return prediction with the selected features provided in Table 4-10 was estimated as $78.86 \pm 0.94\%$. The findings reveal that if we use all of the specified features (given in Table 3-2) for return prediction with the Decision Tree, the accuracy on test data is estimated as 76.5%. The reason for this improvement may be the diversity created by several iterations of the Decision Table with various types of features. Diversity creation as a critical factor in enhancing the accuracy of prediction models is widely discussed in section 4.2.3. Figure 4.4 illustrates the GA trend diagram with the Decision Table fitness function.

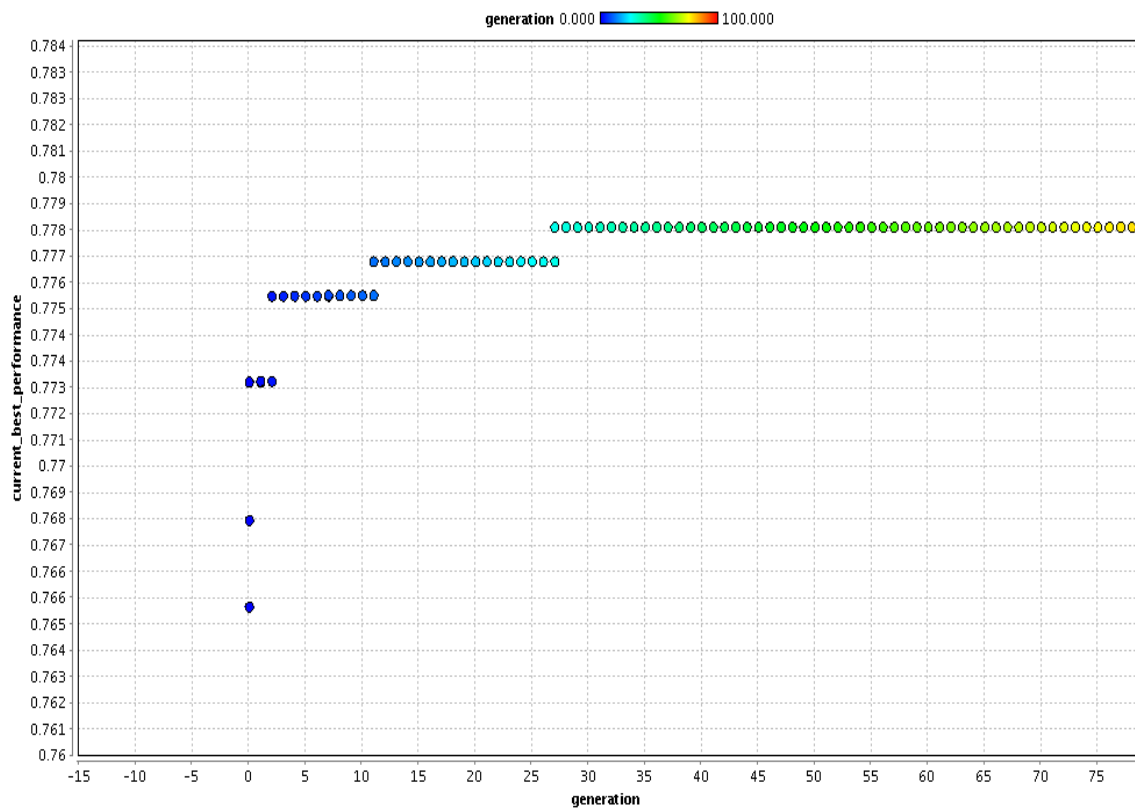


Figure 4.4 Improvement points of GA with Decision Table fitness function for real return prediction

The proposed Wrapper-GA algorithm is also applied for feature selection in risk prediction. Although various fitness and error functions were used in GA, surprisingly, the achievements did not show improvements in prediction accuracy above 70%. Figure 4.5 displays the improvement points of GA for risk prediction when the Decision Table and LAD Tree are used as GA evaluation function and error assessment of selected features, respectively.

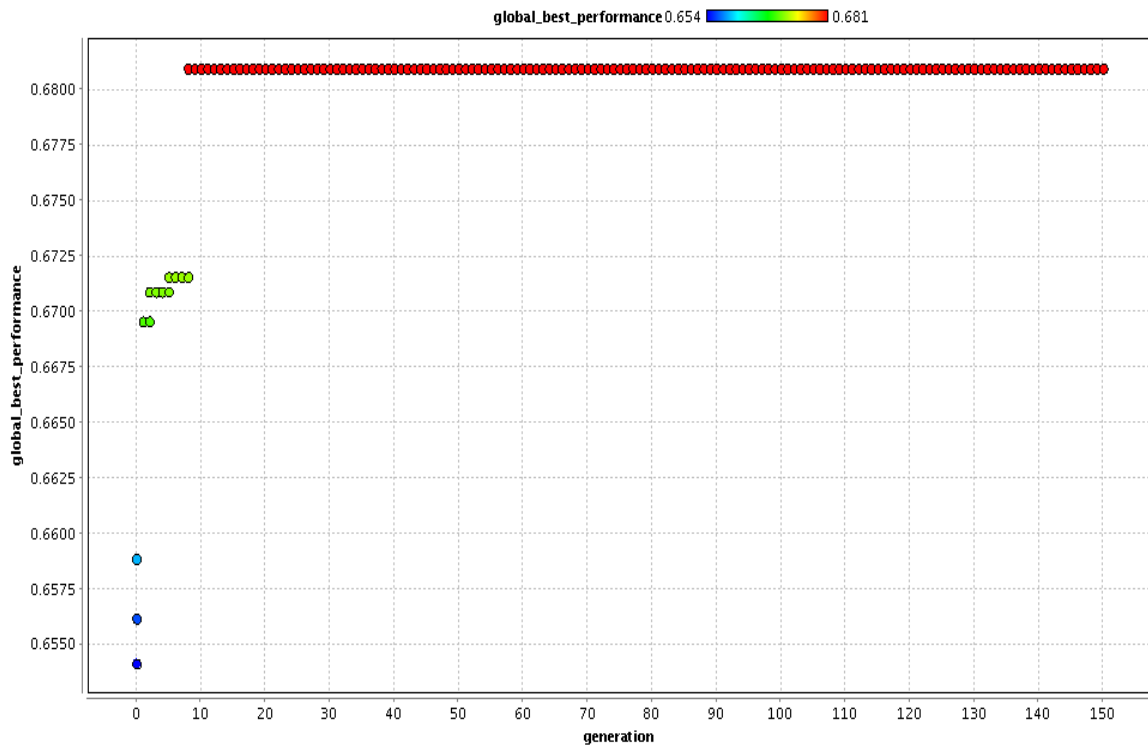


Figure 4.5 Improvement points of GA with LAD Tree fitness function for risk prediction

Table 4-11 provides the selected features with the best accuracy of 68.26 ± 2.55 via Wrapper-GA for risk prediction.

Table 4-11 Selected features for risk prediction with Wrapper-GA algorithm.

Average payment period	Equity ratio	Quick ratio	percentage of net profit to gross profit
Asses the loan usefulness	market return	EPS prediction	β coefficient
P/E	EPS cover percentage	Total income growth percentage	Book value
Efficiency	Debt coverage ratio	Total asset turnover	Fixed asset turnover
current asset turnover	EPS growth percentage	long-term debt to equity ratio	return on equity (ROE)
fixed asset return percentage	current ratio	percentage of net profit to sale	

Although the Wrapper-GA feature selection algorithm could improve the prediction accuracy compared to the case in which all the features are involved, the feature selection ensemble model developed in section 3 outperformed Wrapper-GA in accuracy enhancement. Thus, its related features reported in section 4.4.2 were used for the proposed hybrid prediction method of this chapter.

A comparison between the proposed method and similar studies is shown in Table 4-12. Different hybrid methods that have excellent accuracy in return forecasting in different national stock exchanges were compared based on input data, base classifier, feature selection, hybrid prediction model, and degree of accuracy, as follows:

Table 4-12 Comparison of the proposed fusion scheme versus other studies

Author /Year	Stock Exchange	Input Data	Base Classifier	Feature selection	Hybrid Model	The Best Accuracy%
Tsai, et al. (2011)	Electronic Industry in Taiwan	19 Financial ratios and 11 Macroeconomic indicators	MLP- Cart - Logistic Regression	---	Bagging - Voting	66.67
C.-F. Huang (2012)	30 special companies in Taiwan	14 Financial ratios	SVR- GA	---	---	85- 76.71
J.-H. Cheng, Chen, and Lin (2010b)	Taiwan	10 Technical Indexes and 8 Macroeconomic indicators	PNN- C4.5- Rough Set	---	Hybrid	76
C.-J. Huang, et al. (2008)	South-Korea and Taiwan	23 Technical Indexes	SVM- K-NN- Cart- Logistic Regression- Back Propagation	Wrapper	Voting	76.06 80.28
Tsai and Hsiao (2010)	Taiwan	8 Fundamental Index and 11 Macroeconomic indicators	---	GA-PCA-Cart	Back Propagation	79
Tsai, Lu, and Yen (2012)	Taiwan	61 intangible assets value variable	MLP	PCA- Stepwise Regression- decision trees- association rules- GA	MLP	75
Barak and Modarres (2015)	Return Forecasting in TSE-Iran	44 Financial ratios and Fundamental Index	Cart, Rep Tree, LAD Tree, ...	Function based Clustering	Hybrid	80.24
Barak and Modarres (2015)	Risk Forecasting in TSE-Iran	44 Financial ratios and Fundamental Index	DTNB, BF Tree, LAD Tree, ...	Function based clustering	Hybrid	79.01
Current chapter	Return Forecasting in TSE-Iran	44 Financial ratios and Fundamental Index	Cart, Rep Tree, LAD Tree, ...	Function-based clustering with diversity	Fusion	83.65
Current chapter	Risk Forecasting in TSE-Iran	44 Financial ratios and Fundamental Index	DTNB, BF Tree, LAD Tree, ...	Function-based clustering with diversity	Fusion	88.23

4.6.Conclusions and Future Research

In this chapter, a study of fusion models based on the use of multiple diversity classifiers is presented for stock return and risk prediction. Bagging, Boosting, and AdaBoost were applied as three diversity algorithms for generating a pool of classifiers. An empirical study was later undertaken on the Tehran Stock Exchange that compared the performance of diversity algorithms with different sets of classifiers in a fusion system. Bagging consistently outperformed the other two algorithms, regardless of the type of individual classifiers employed.

Almost all of the fusion strategies provided statistically significant improvements in performance over the best individual classifiers. Bagging performed well with Decision Trees in fusion systems that are stated as being unstable prediction methods. The limitation of this method is that collecting all of the fundamental data and information may be difficult for certain cases.

Future research directions of the chapter include but are not limited to:

1. Optimizing the parameters of classification algorithms using meta-heuristics algorithms to improve the prediction results;
2. Predicting other important response variables (in addition to risk and return) such as liquidity (Barak, et al., 2013);
3. Using technical features and textual information, in addition to fundamentals features, in order to use more comprehensive features and be able to predict the short term situations of stocks; and
4. Customizing the proposed approach for the prediction of risk and return in a particular industry or investigating the accuracy of the procedure using data from other popular stock markets, such as the US stock market, which may result in new dimensions for this procedure.

Chapter 5

Wrapper ANFIS-ICA method to do stock market timing and feature selection based on Japanese Candlestick

Summary of the fifth chapter

This chapter presents a novel forecasting model for stock markets based on the wrapper ANFIS (Adaptive Neural Fuzzy Inference System) – ICA (Imperialist Competitive Algorithm) and technical analysis of Japanese Candlestick. Two approaches of Raw-based and Signal-based are devised to extract the model's input variables with 15 and 24 features, respectively. The correct predictions percentages for periods of 1-6 days with the total number of buy and sell signals are considered as output variables. In the proposed model, the ANFIS prediction results are used as a cost function of the wrapper model, and ICA is used to select the most appropriate features. This novel combination of feature selection not only takes advantage of ICA optimization swiftness but also the ANFIS prediction accuracy. The emitted buy and sell signals of the model revealed that Signal databases approach gets better results with 87% prediction accuracy and the wrapper features selection obtains 12% improvement in predictive performance regarding the base study. Additionally, since the wrapper-based feature selection models are considerably more time-consuming, our presented wrapper ANFIS-ICA algorithm's results have superiority in time decreasing as well as prediction accuracy increasing regarding other algorithms such as wrapper Genetic algorithm (GA).

5.1.Introduction

A good prediction contributes to better decision-making and planning for the future while it allows the current variables to create considerable value. Predictions accuracy affects the organization's financial stability (Asadi, et al., 2012). In the industry of information processing, data mining is an ever-growing technology and has been applied in engineering, business, and management. Prediction of stock prices, credit scores, and even bankruptcy potentials are examples of significant applicability of data mining in the field of finance. Since financial markets are complex and non-linear dynamic systems, their predictions are challenging (C.-L. Huang & Tsai, 2009). There are different forecasting approaches such as statistical models, technical analysis (TA), and econometric methods (Atsalakis, et al., 2011; Kar, et al., 2014) while in this study the focus is on wrapper approach based on TA.

The most crucial points for trading are peaks and bottoms of the price trend in the way that a professional and lucky practitioner sells at peaks and buys at bottoms. In the real world, TA gives considerable assistance

in such decisions (X. Lin, Yang, & Song, 2011) and its application has been well conceived by the financial market experts (Menkhoff, 2010). It should be noted that for short term period prediction, TA is the best tool although lately, it has gone beyond the intention of short-term trading (Cervelló-Royo, Guijarro, & Michniuk, 2015; Yingzi Zhu & Zhou, 2009).

Due to the discussed importance, over the last two decades, many researches have focused on the development of intelligent soft computing models (Bisoi & Dash, 2014; Hafezi, Shahrabi, & Hadavandi, 2015; Majhi, Panda, & Sahoo, 2009). Most of them combine soft computing techniques and TA with stock analysis such as (Y. Chen, Mabu, Shimada, & Hirasawa, 2009; Patel, Shah, Thakkar, & Kotecha, 2015b; Wen, Yang, Song, & Jia, 2010).

Since the end of the 1980s, some financial scholars applied artificial neural networks (ANNs) to predict the stock market (K. Lee & Jo, 1999). However, the ANNs have some shortcomings in patterns learning because stock market data are noisy and with complex dimensions (C.-J. Huang, et al., 2008). Therefore, to deal with these challenges, scholars proposed a combination of fuzzy theory and ANNs.

A Takagi–Sugeno–Kang-Type Neuro-fuzzy rule-based system to forecast Taiwan Stock Exchange price deviation is developed by Chang and Liu (P.-C. Chang & Liu, 2008) with an accuracy of 97.6% and 98.08% in TSE index and MediaTek respectively. A hybrid Neuro-fuzzy with ANFIS (Adaptive Neural Fuzzy Inference System) is developed by Yunos et al. (Yunos, Shamsuddin, & Sallehuddin, 2008) to predict daily movements of the Kuala Lumpur Composite Index (KLCI) and the results show the superiority of the ANFIS to ANN. In comparison with the others, ANFIS has a high speed of training, the most effective learning algorithm, and is simple in terms of the structure (Sarkheyli, Zain, & Sharif, 2015). ANFIS provides better results when applied without any pre-training (Vairappan, Tamura, Gao, & Tang, 2009). (Bagheri, Mohammadi Peyhani, & Akbari, 2014) used an ANFIS with QPSO (Quantum-behaved Particle Swarm Optimization) hybrid method for financial forecasting. For the scholars from this field, the review paper of Atsalakis and Valavanis (Atsalakis & Valavanis, 2009) and Kar, et al. (2014) can be useful.

Technical knowledge usually is elicited by TAs, and in this study, a Japanese Candlestick chart analysis is used in order to elicit technical knowledge. Although the supporting literature on the Japanese Candlestick charting and its antiquity are important reasons for the application of them in decision making, but the two following points are the main points for its application in our new model:

1. Consideration of open, high, low and close prices, unlike the other works that use only close prices.
2. Being more robust than the other technical trading rules from the perspective of data snooping (Jasemi, Kimiagari, & Memariani, 2011).

Since there are many indices in the stock exchange and more specifically in TA problems, feature selection becomes much more important. A number of studies have claimed and verified that feature selection (FS) is the key process in stock prediction (Barak & Modarres, 2015; Tsai & Hsiao, 2010). As mentioned in the

past chapters feature selection decreases the calculation cost by decreasing the corresponding dimensionality or improves the forecasting performance by the elimination of extra and unrelated features (Crone & Kourentzes, 2010). Generally there are 2 types of feature selection methods which are: (i) Wrapper methods, and (ii) Filter methods (Barak & Modarres, 2015; Y.-S. Chen & Cheng, 2012) while wrapper algorithms are more successful in most cases (Chiang, Urban, & Baldrige, 1996; Kohavi & John, 1997; Lawrence, Giles, & Tsoi, 1997; S. Maldonado & R. Weber, 2009; Min & Lee, 2005; A. Wang, An, Chen, Li, & Alterovitz, 2015).

To sum up, in this chapter, a novel hybrid feature selection algorithm based on the wrapper ICA-ANFIS method is applied to select the important features. What makes our proposed approach different from the previous ones are that we proposed a new combination of Japanese Candlestick charts and wrapper ANFIS-ICA for forecasting.

Application of Japanese Candlestick charts as technical analysis is a key advantage of this chapter and the effectiveness of the selected features of this pattern is proved. Moreover, in this chapter, the applied wrapper is a new combination in the way that ANFIS is an analyzer to predict future of the stock market trend, and feature selection is done by ICA (Imperialist Competitive Algorithm). The model not only increases the prediction accuracy but also the processing pace and finally, it should be noted that the patterns of Japanese Candlestick are on the basis of Jasemi, et al. (2011).

The contribution of the chapter is summarized as follows:

- Gathering a comprehensive Japanese Candlestick database regarding new adaptive technical analysis.
- Developing a hybrid feature selection algorithm based on the wrapper ICA-ANFIS model.
- Exploring how much efficient an ANFIS would be to infer the TA clues if it is provided with important indices of the TA (based on wrapper feature selection) and actual trends.

The rest of Chapter 5 is organized as follows: section 5.2 reviews the literature of wrapper, ANFIS, and ICA; section 5.3 introduces the model; section 5.4 presents the experimental results, and finally, section 5.5 focuses on the conclusions.

5.2. The background

5.2.1. Wrapper feature selection method

In the Wrapper method, the goal is to find a subset of size r from n variables ($r < n$) that maximizes the predictor performance (S. Maldonado & R. Weber, 2009). The method utilizes the learning mechanism as the fitness function and seeks the best subset of the features while standard optimization techniques with learning mechanisms for ranking of the subsets are possible. Kohavi and John (Kohavi & John, 1997) have a leading role in the popularization of the wrapper approach that is really powerful in feature selection, but it has its computational complexities, and it is more time consuming than Filter method (J. Huang, Cai, & Xu, 2007). Figure 5.1 shows the approach concept.

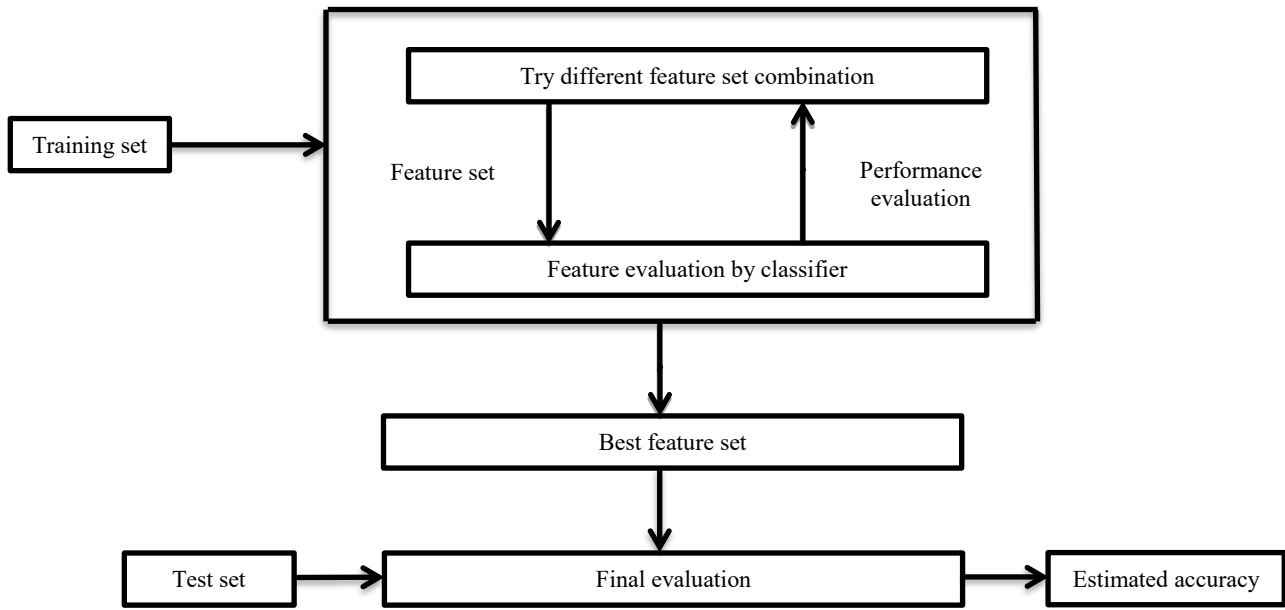


Figure 5.1 The approach of Wrapper

For a given dataset of G with N features, the wrapper approach starts from a subset of F_0 (an empty set, a full set, or any randomly selected set) and with a particular strategy searches the features space. It evaluates each generated subset of F_i by applying a learning model that considers F_i as an input, and if the learning model performance improves with F_i , F_i is regarded as the best current subset. Then the wrapper modifies F_i by adding or eliminating features and performing the search until coming to the predefined stopping criterion (Kabir & Islam, 2010). The above-mentioned feature searching problem is NP-hard, and the number of local minima can be quite large and naturally a wide range of heuristic search strategies including forwarding selection, backward elimination, hill-climbing, branch, and bound algorithms, and meta-heuristics algorithms like simulated annealing and genetic algorithms (GAs) have been used (J. Huang, et al., 2007).

5.2.2. Adaptive Neural Fuzzy Inference System (ANFIS)

ANFIS is a multi-layer adaptive network-based fuzzy inference system proposed by Jang (Jang, 1993). An ANFIS consists of totally five layers to implement different node functions to learn and tune parameters in a Takagi–Sugeno fuzzy inference system (FIS) using a hybrid learning model.

The first layer executes a fuzzification process, the second layer executes the fuzzy AND of the antecedent part of the fuzzy rules, the third layer normalizes the membership functions, the fourth layer executes the conclusion part of the fuzzy rules, and the last layer computes the output of the fuzzy system by summing up the outputs of the four layers. The feed forward equations of the ANFIS structure with two inputs and two labels for each input are presented as follow:

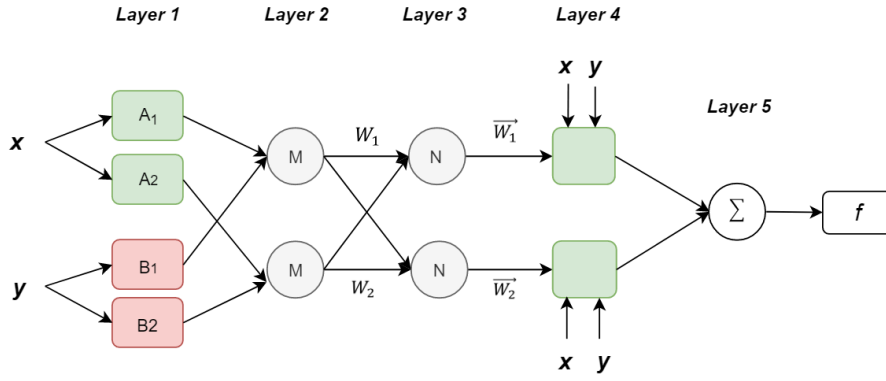


Figure 5.2 ANFIS architecture

Layer 1:

$$O_i^1 = \mu_{A_i}(x), i = 1,2 \quad \text{Eq. 5-1}$$

$$O_i^1 = \mu_{B_{i-2}}(y), i = 3,4 \quad \text{Eq. 5-2}$$

Layer 2:

$$O_i^2 = w_i = \mu_{A_i}(x) * \mu_{B_i}(y), i = 1,2 \quad \text{Eq. 5-3}$$

Layer 3:

$$O_i^3 = \bar{w}_i = \frac{w_i}{w_1 + w_2}, i = 1,2 \quad \text{Eq. 5-4}$$

Layer 4:

$$O_i^4 = \bar{w}_i \cdot f_i = \bar{w}_i \cdot (p_i x + q_i y + r_i), i = 1,2 \quad \text{Eq. 5-5}$$

Layer 5:

$$O_i^5 = \sum_{i=1}^2 \bar{w}_i \cdot f_i \quad \text{Eq. 5-6}$$

where x and y are inputs to node i , A_i and B_i are linguistic labels for inputs, and $\{p_i, q_i, r_i\}$ are the parameters setting.

ANFIS model design consists of two sections: constructing and training. In the construction section, the number and type of FIS structure are defined. Construction of the ANFIS model requires the division of the input/output data into rule patches. This can be achieved by using several methods such as grid partitioning, subtractive clustering method and fuzzy c-means (FCM).

When there is only one output, FCM can be applied for making a primary FIS to train the ANFIS (Bezdek, 1981). FCM is done with minimizing a goal function that represents the distance of each data point to the data center that has been weighted by a membership degree of the data point (Eq. 4).

$$\min J_m(U, V) = \sum_{j=1}^n \sum_{i=1}^c U_{ij}^m \|x_i - v_j\| \quad \text{Eq. 5-7}$$

Where N is the total number of patten in the data set, m is a real number greater than one, $x_i (\in R_p)$ is the i -th feature data, v_j is the center of the j -th cluster and U_{ij} shows the degree of membership of x_i in the j -th cluster, and, and $\|*\|$ is any norm that shows the distance between each measured data and the center. With iterative optimization of the above objective function, fuzzy partitioning is done, by updating membership U_{ij} and cluster centers v_j as follow:

$$V_j = \frac{\sum_{j=1}^n U_{ij}^m x_j}{\sum_{j=1}^n U_{ij}^m} \quad \text{Eq. 5-8}$$

$$U_{ij} = \left(\sum_{k=1}^c \left(\frac{\|x_j - v_i\|}{\|x_j - v_k\|} \right)^{\frac{1}{m-1}} \right)^{-1}, \quad 1 \leq i \leq c, \quad 1 \leq j \leq n \quad \text{Eq. 5-9}$$

Where k is the step's number of iteration and when there is no improvement in $J_m(U, V)$, the iteration will stop (Esfahanipour & Aghamiri, 2010) (Sarkheyli, et al., 2015).

5.2.3. Imperialist Competitive Algorithm (ICA)

ICA is a new socio-politically motivated global search strategy that has been introduced for dealing with different optimization tasks (Atashpaz-Gargari & Lucas, 2007). This evolutionary optimization strategy has shown great performance in both convergence rate and better global optima achievement (Atashpaz-Gargari & Lucas, 2007; Biabangard-Oskouyi, Atashpaz-Gargari, Soltani, & Lucas, 2008; Gargari, Hashemzadeh, Rajabioun, & Lucas, 2008; Khoshnevisan, et al., 2015; Rajabioun, Atashpaz-Gargari, & Lucas, 2008; Sepehri Rad & Lucas, 2008). Nevertheless, its effectiveness, limitations, and applicability in various domains are currently extensively investigated.

Figure 5.3 (Atashpaz-Gargari & Lucas, 2007) shows the flowchart of the ICA. Similar to other evolutionary algorithms, this algorithm starts with an initial population. Each of the population is called a country. Some of the best countries (in optimization terminology, countries with the least cost) are selected to be the imperialist states and the rest form the colonies of these imperialists. All the colonies of initial countries are divided among the mentioned imperialists based on their power. The power of each country, the counterpart of fitness value in the GA, is inversely proportional to its cost. Then an imperialist with its colonies can found an empire. Figure 5.4 shows the primary empires while the most powerful imperialist 1 has more countries than the others.

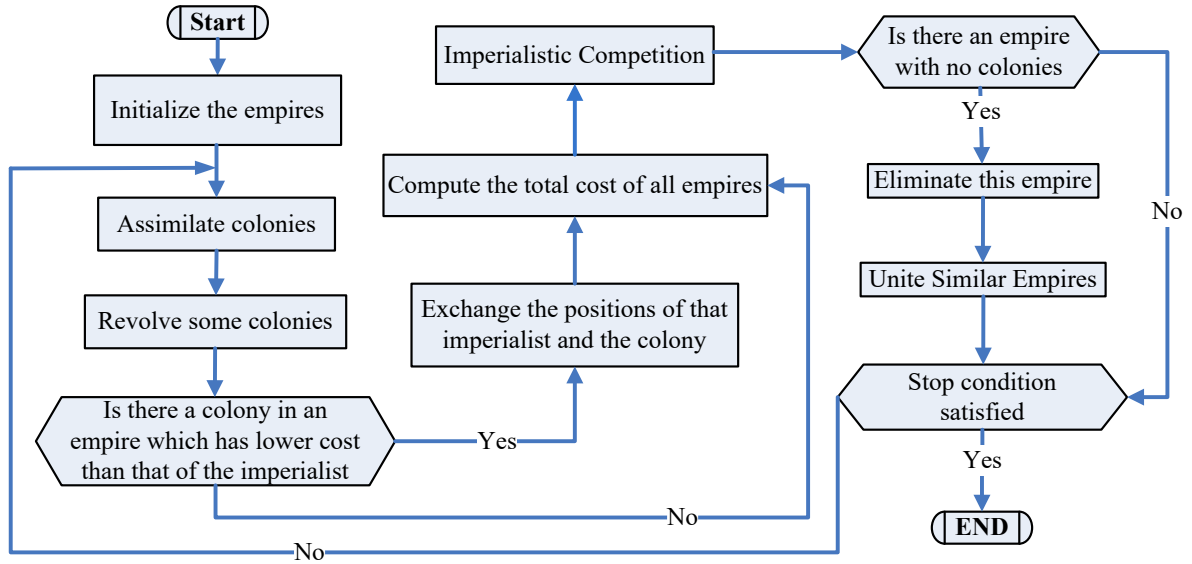


Figure 5.3 Flowchart of the Imperialist Competitive Algorithm

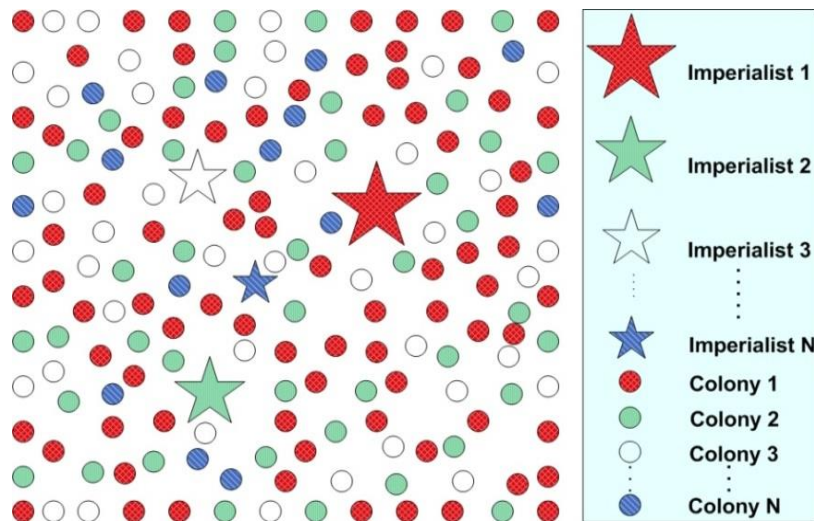


Figure 5.4 Making the primary empires

After forming initial empires, the colonies in each of them start moving toward their relevant imperialist country. This movement is a simple model of assimilation policy which was pursued by some of the imperialist states. The Assimilation makes the colonies of each empire in searching space, closer to the imperialist position while the revolution covers random changes in the position of some of the countries. During Assimilation and revolution, a colony may advance to a better position and has the probability of controlling the entire empire and replace the imperialist.

If after this move, one of the colonies has more power than the imperialist, they will be replaced. To initiate the competition among the empires, the objective function of each empire is calculated. This function depends on the imperialist objective function and its colonies. Then the competition starts and the weakest empire lose its assets, and the powerful empire tries to achieve them. The empire that loses all of its colonies

will vanish. Finally, the most powerful empire possesses the other empires assets and wins the competition. From the optimization point of view, this leads to finding the optimal solution of the problem, i.e. solution with least cost value.

5.3.The proposed model

5.3.1. Adaptive TAs and ANFIS

Technical analysis normally has two general approaches to evaluate the stock prices while the first uses technical indicators and oscillators, and the second uses charts (Chavarnakul & Enke, 2008). Whether mathematical or pattern charts, the mechanism of a typical TA is based on signs and (their associated) signals, while signs are generated by the stock price alteration and signals are restricted to three states of ascending, descending and neutral (Lu, 2014).

The concept of Adaptive TAs is not much different from the old one. The new generation of TA models is again categorized into two categories of mathematical and pattern charts and it is again a system of signs and signals but unlike the old static one, it is dynamic.

In traditional combination, target data are results of applying the input data to some pre-determined rules of the selected TA. That is the rules are constant and the concern of model developer is to better educate the network. In traditional combination, the target data is determined according to the input data and the predetermined rules. In statistic words, the independent and dependent variables are input and target data, respectively. The following rules (Rule 1-6) are some of the predetermined rules in TA which are illustrated in Figure 5.5 (Lu, Shiu, & Liu, 2012). The focus is on open (O_i), high (H_i), low (L_i) and close (C_i) prices of the stock in the i^{th} day due to the Japanese Candlestick (do Prado, Ferneda, Morais, Luiz, & Matsura, 2013).

1. The Piercing: $O_1 > C_1$, $O_2 < C_2$, $O_2 \leq C_1$, $C_2 < O_1$, and $C_2 > C_1 + 0.5 (O_1 - C_1)$.
2. The Bullish Engulfing: $O_1 > C_1$, $O_2 < C_2$, $O_2 \leq C_1$, and $C_2 \geq O_1$.
3. The Bullish Harami: $O_1 > C_1$, $O_2 < C_2$, $O_2 > C_1$, and $C_2 < O_1$.
4. The Dark-cloud Cover: $O_1 < C_1$, $O_2 > C_2$, $O_2 \geq C_1$, and $C_2 < C_1 - 0.5 (C_1 - O_1)$.
5. The Bearish Engulfing: $O_1 < C_1$, $O_2 > C_2$, $O_2 \geq C_1$, and $C_2 \leq O_1$.
6. The Bearish Harami: $O_1 < C_1$, $O_2 > C_2$, $O_2 < C_1$, and $C_2 > O_1$.

As we can see in the upper rules, they are some predetermined rules; however, in this present chapter, the approach are completely dependent on input and target data in the way that they may be completely different for two distinct periods of time. In other words, there is no pre-determined rule in the proposed synthetic model but only some inputs and their associated target data based on which, new rules are extracted. The most important task in the new model is with the analyzer that confirms the old rules or extracts the new

ones. Figure 5.6 shows how the analyzer relates input data, target data, and the new technical rules together. Also the new approach inserts some kind of continuity to the selected rules.

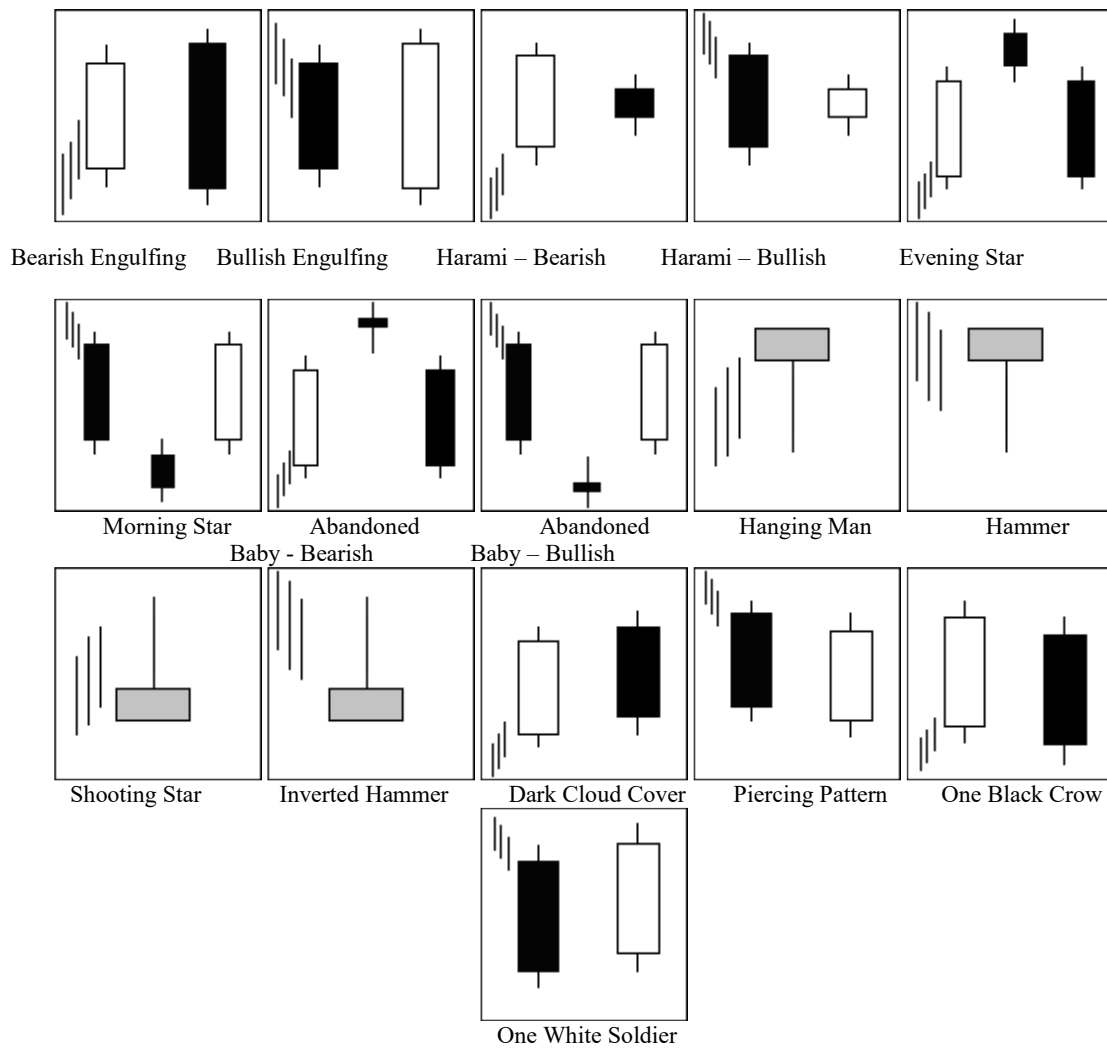


Figure 5.5 Traditional 2-days patterns



Figure 5.6 The new relation of input data and technical analysis rules

In this study, an ANFIS like in many similar works in the literature plays the analyzer role. However, unlike the previous works, in the new model, the ANFIS knowledge changes over time according to new training data.

Therefore in this study, a modern combination of Japanese Candlestick charting and ANFIS is presented. It is explored how much efficient an ANFIS would be to infer the TA clues if it is provided with important indices of the TA (based on wrapper feature selection) and actual trends. This model created for presented research, if successful, could be beneficial for stock traders, much in allowing them to make better decision making using the most up-to-date investment techniques.

5.3.2. Model

The model proposes an adaptive TA based novel methodology for signal prediction and feature selection with a wrapper approach called ANFIS-ICA, which results from the combination of ANFIS as a signal analyzer and ICA, as feature finder and subset evaluator. In other words, this model uses ANFIS as a signal analyzer tool while a novel wrapper feature selection algorithm selects important Candlestick based features to feed the model. Figure 5.7 shows the general structure of the model while it will be discussed in details in the following parts.

The following parameters are used throughout the study:

N_{attrib}: Number of features

N_{pop}: Number of initial population

N_{imp}: Number of imperialists

Zeta: The effect coefficient of the colonies cost on the empire cost

P_{revolution}: Probability of revolution

Imp Colonies Costs (i): Colonies costs of i^{th} imperialist

Imp costs (i): Cost of i^{th} imperialist

Imp fitness (i) : Fitness of i^{th} imperialist

Max Decades: Maximum periods as stopping criterion in the ICA

Num MFs: Number of membership functions for FCM

Epoach_n: Number of train epochs in the FCM

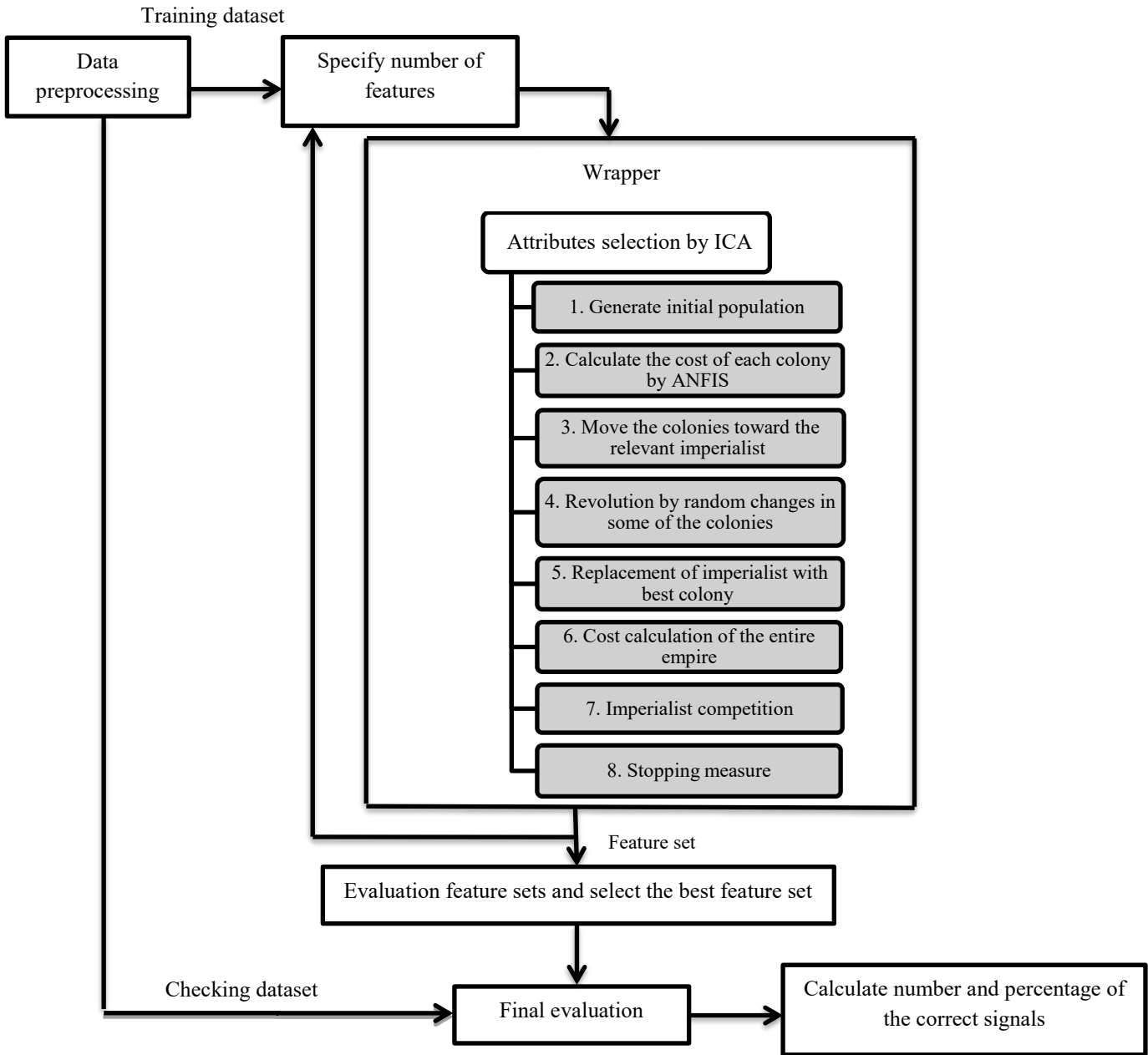


Figure 5.7 The general structure of the presented model

- **Development of the initial population in ICA**

To generate the initial population, N_{Pop} random permutation of integers from 1 to N_{Attrib} in a matrix structure $(N_{Pop} \times N_{Attrib})$ as it is shown in Figure 5.8 is developed (the figure is only an example for a possible population).

	1	.	.	.	N_{attrib}	
1	2	7	5	9...	14	→ Country
.	3	13	2	10...	5	
.	11	6	2	1...	8	
.	8	6	1	13...	15	
N_{pop}	7	14	9	2...	6	

Figure 5.8 Artificial example of making the primary population

The colonies are organized on the basis of the initial population. The i -th population with a position that includes N_{attrib} features makes the i -th colony, while j -th number in the colony is the number of the feature. Then the cost of each colony is calculated, and the colonies are sorted based on their cost ascend, and the most powerful colonies are selected as imperialists, and the others ($N_{pop} - N_{imp}$) are colonies of these imperialists [see (Atashpaz-Gargari & Lucas, 2007)].

- **Cost function**

ANFIS model is applied to calculate the cost function in ICA. Firstly the FCM function is applied for appointing the number of rules and membership functions. FCM uses Gaussian and Linear functions for input and output membership functions respectively. After making the initial FIS structure, ANFIS function is applied to train the system by the initial FIS structure that is made by FCM and training data. Then training $RMSE$ of fuzzy output is calculated by Eq. 5-10.

$$RMSE = \sqrt{\frac{\sum_{t=1}^n (y'_t - y_t)^2}{n}} \quad \text{Eq. 5-10}$$

while, y'_t is the result of prediction and y_t is the real amount of t -th data.

After the training phase, checking dataset is used to check the ANFIS. Furthermore, checking $RMSE$ is calculated. The check $RMSE$ is used as the performance measure and cost function in ICA.

- **Assimilation policy**

To draw the colony closer to the imperialist (Assimilation), 2 random numbers from 1 to N_{attrib} are selected as R_1 and R_2 , and then C_1 and C_2 are considered as minimum and maximum of them respectively. Then from 1 to C_1-1 features of the imperialist are considered as 1 to C_1-1 features of the child country; from C_1 to C_2 features of colony are considered as C_1 to C_2 features of the child country; from C_2+1 to n features of the imperialist are considered as C_2+1 to n features of the child country and so the features of the child country is achieved. Figure 5.9 shows an example of Assimilation with $R_1=3$ and $R_2=6$; $C_1=3$ and $C_2=6$.

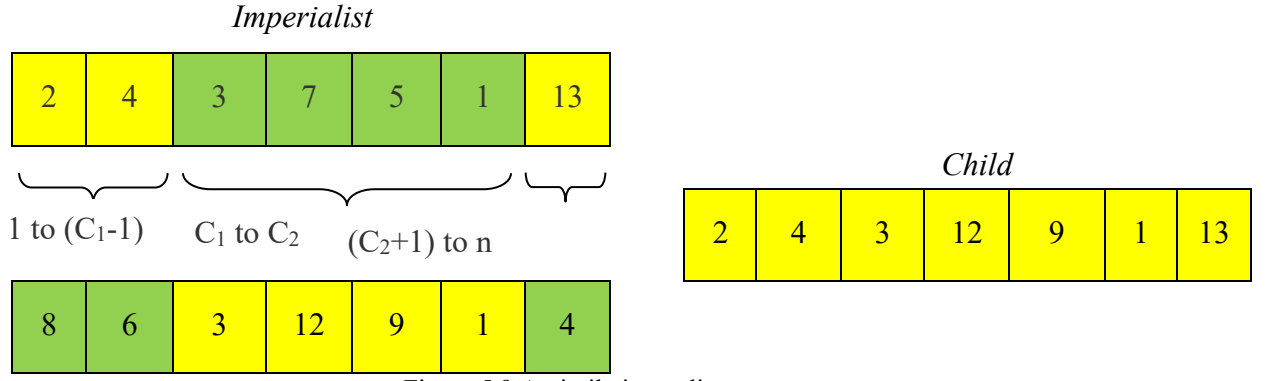


Figure 5.9 Assimilation policy

The new country (child) is a combination of the previous colony that converges to the imperialist country.

- **Revolution**

For the j -th colony of the i -th imperialist if revolution chance (random number) is smaller than P , revolution is done by random changes in some of the colonies. In this regard, two random numbers of k_1 and k_2 from 1 to N_{attrib} are selected and then feature k_1 from the j -th colony, and the i -th empire is replaced with feature k_2 from that exact colony, and so the revolution function in some of the countries is done. Figure 5.10 shows an example of the revolution.

\longleftrightarrow

Before	8	6	3	12	9	1	4
After	8	6	9	12	3	1	4

Figure 5.10 Revolution

- **Replacement of the imperialist with the best colony**

If one of the colonies possesses more power than its relevant imperialist, they will exchange their positions.

- **Cost calculation of the entire empire**

The cost of each empire depends on the costs of imperialist and colonies as follows:

$$Imp\ fitness(i) = Imp\ costs(i) + zeta * \frac{\sum_{t=1}^n Imp\ Colonies\ Costs(t)}{n} \quad i = 1, 2, \dots, N_{imp} \quad Eq. 5-11$$

where n is the number of colonies in i^{th} empire.

- **Imperialist competition**

In this regard, first of all the empire fitness is updated by Eq. 5-12 and then the intended probability is got by Eq. 5-13.

$$Update\ Imp\ fitness(i) = \max_1^{N_{imp}} Imp\ fitness(n) - Imp\ fitness(i) \quad i = 1, \dots, N_{imp} \quad Eq. 5-12$$

$$p_i = \frac{\text{update Imp fitness } (i)}{\sum_{n=1}^{N_{imp}} (\text{Imp fitness } (n))} \quad i = 1, 2, \dots, N_{imp} \quad \text{Eq. 5-13}$$

After calculating the probabilities, the vector of \mathbf{P} is organized as follows:

$$\mathbf{P} = [p_1, p_2, p_3, \dots, p_{N_{imp}}] \quad \text{Eq. 5-14}$$

To distribute the colonies among the empires according to their possession probabilities, the approach of roulette is applied. The Empire with more fitness has higher probability to be selected. After this stage, it should be checked that if the weakest empire does not have any colony, and then it must be allocated as a colony to other empires based on fitness probability of the empires and roulette wheel.

- **Stopping measure**

The algorithm is repeated until it comes to a pre-determined number of repetitions and finally, the best answer is selected from them.

To sum up these steps, the semi-codes of the model are as follows:

- (1) Set up ICA Settings:
 - *Max Decades*; N_{pop} ; N_{imp} ; $\mathbf{P}_{revolution}$; \mathbf{zeta} ;
 - $N_{col} = N_{pop} - N_{imp}$;
- (2) Set up training and checking data and $\mathbf{epoch_n}$
- (3) $\theta = 1$
- (4) Generate the initial population
- (5) Calculate the cost of each colony:
 1. Develop the FIS structure by FCM.
 2. Train the ANFIS by the initial FIS structure.
 3. Validate the model by $RMSE$ and compute the train $RMSE$.
 4. Validate the model by the checking data.
 5. Compute the check $RMSE$ and set as cost function.
- (6) Select the most powerful colonies as imperialists.
- (7) Allocate rest of the colonies to the imperialist on the basis of the imperialist power.
- (8) Generate empires by imperialists and their colonies.
- (9) Move the colonies toward the relevant imperialist.
- (10) Revolve in some of the colonies.
- (11) Exchange the position of a colony and the imperialist if its cost is lower.
- (12) Compute the objective function (the total cost) of all empires.
- (13) Pick the weakest colony and give it to the best empire.
- (14) Eliminate the powerless empires.

- (15) If the stop condition is satisfied stop, and if not go to step 10.
- (16) Select the optimum subset of the features with the minimum check error (Check RMSE).
- (17) If $\theta = \text{Max decades}$ then stop, and if not $\theta = \theta + 1$ and go to step 5.

5.4. Empirical results

5.4.1. Input data

In this study, the training data are based on the two applied approaches of Jasemi et al. (Jasemi, et al., 2011). The first approach (Raw database) is based on Raw input features including 15 items and 1 output. In this approach the focus is on open (O_i), high (H_i), low (L_i) and close (C_i) prices of the stock in the i^{th} day due to the Japanese Candlestick during last 3 days while to cover the stock price trend the close prices of the stock during the last 7 days are also included. Totally this approach comes to 15 normalized indices of $\frac{C_i}{C_1}$ $i=2,3,4$; $\frac{O_i}{C_1}$, $\frac{H_i}{C_1}$, $\frac{L_i}{C_1}$ and $\frac{C_i}{C_1}$ $i=5,6,7$.

The 15 indices of this approach are shown in Table 5-1.

Table 5-1 Raw approach

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$\frac{C_2}{C_1}$	$\frac{C_3}{C_1}$	$\frac{C_4}{C_1}$	$\frac{C_5}{C_1}$	$\frac{C_6}{C_1}$	$\frac{C_7}{C_1}$	$\frac{O_5}{C_1}$	$\frac{H_5}{C_1}$	$\frac{L_5}{C_1}$	$\frac{O_6}{C_1}$	$\frac{H_6}{C_1}$	$\frac{L_6}{C_1}$	$\frac{O_7}{C_1}$	$\frac{H_7}{C_1}$	$\frac{O_7}{C_1}$

The second approach (Signal database) is based on the reverse signals of Japanese Candlestick technique including 24 input features and 1 output. This package covers the important factors of decision making in the technique. In this approach, the focus is on the reversal signals of Japanese Candlestick like Morning star, Inverted hammer, Harami, and many others. This set of inputs is more advanced than the previous and covers important pattern charts' clues of the TA which are defined as shown by Table 5-2.

Table 5-2 Signal approach

1	2	3	4	5	6	7	8	9	10	11	12
$\frac{C_2}{C_1}$	$\frac{C_3}{C_1}$	$\frac{C_4}{C_1}$	$\frac{C_5}{C_1}$	$\frac{C_6}{C_1}$	$\frac{C_7}{C_1}$	$\frac{O_5}{C_5}$	$\frac{O_6}{C_6}$	$\frac{O_7}{C_7}$	$\frac{H_7}{\text{Max}(O_7, C_7)}$	$\frac{\text{Min}(O_7, C_7)}{L_7}$	$\frac{\text{Max}(O_7, C_7)}{\text{Max}(O_6, C_6)}$
13	14	15	16	17	18	19	20	21	22	23	24
$\frac{\text{Min}(O_7, C_7)}{\text{Min}(O_6, C_6)}$	$\frac{O_7}{H_6}$	$\frac{L_6}{O_7}$	$\frac{C_7}{O_6}$	$\frac{\text{Max}(O_6, C_6)}{\text{Min}(O_5, C_5)}$	$\frac{\text{Max}(O_7, C_7)}{\text{Max}(O_5, C_5)}$	$\frac{\text{Min}(O_7, C_7)}{\text{Min}(O_5, C_5)}$	$\frac{\text{Min}(O_6, C_6)}{\text{Max}(O_5, C_5)}$	$\frac{H_7}{H_6}$	$\frac{H_7}{H_5}$	$\frac{L_6}{L_5}$	$\frac{L_7}{L_5}$

These approaches in 48 data sets, according to Table 5-3, are applied to train and test the introduced ANFIS model. This structure gains the advantage of adaptive TA in which the input data and the target data are dependent and change over time. Also, this structure is enjoyed from sliding window function in time series prediction (Mozaffari, Mozaffari, & Azad).

Table 5-3 Details of the applied data sets

NO.	Period of training data(year)	Period of testing data(year)	NO.	Period of training data(year)	Period of testing data(year)	NO.	Period of training data(year)	Period of testing data(year)	NO.	Period of training data(year)	Period of testing data(year)
1	2000	2001	13	2000-2001	2006	25	2000-2003	2007	37	2001-2002	2006
2		2002	14		2007	26		2008	38		2007
3		2003	15		2008	27	2001	2002	39		2008
4		2004	16	2000-2002	2003	28		2003	40	2001-2003	2004
5		2005	17		2004	29		2004	41		2005
6		2006	18		2005	30		2005	42		2006
7		2007	19		2006	31		2006	43		2007
8		2008	20		2007	32		2007	44		2008
9	2000-2001	2002	21		2008	33		2008	45	2001-2004	2005
10		2003	22	2000-2003	2004	34	2001-2002	2003	46		2006
11		2004	23		2005	35		2004	47		2007
12		2005	24		2006	36		2005	48		2008

5.4.2. Results

The input data of our experiment belong to daily stock prices of General Motors Company at New York Stock Exchange from 2000 to 2009. 48 data sets according to Table 5.3 are applied for learning and checking.

To optimize the ANFIS parameters, the pre-assumed hybrid method of pre-propagation and least squares is applied. After different evaluation, 20 membership functions were selected as the optimal number of FCM rules. In ICA based on different runs, the total number of colonies, number of empires, rate of colonies revolution (P), effect coefficient of colonies power on empire power (ζ) and maximum number of repetition as the stop condition are respectively 20, 5, 0.5, 0.1 and 30.

The following 4 figures show the outputs of the wrapper ANFIS-ICA model with the raw input data of 2000-2001 for train and 2003 for checking. Figure 5.11.a shows the features attitudes of the data approach. The upper trend shows the features trend, and the bottom represents the outputs for one sample of normalized data. Figure 5.11.b shows the empires costs and the stages of imperials elimination in ICA in the way that the weak imperials are eliminated and the strong ones remain. Figure 5.11.c shows the decrease of $RMSE$ (cost) in repetition of the algorithm and represents two charts of the costs mean and the minimum cost in each repetition. Figure 5.11.d shows checking the output and ANFIS output with the root of mean square error for the best position on top of the figure.

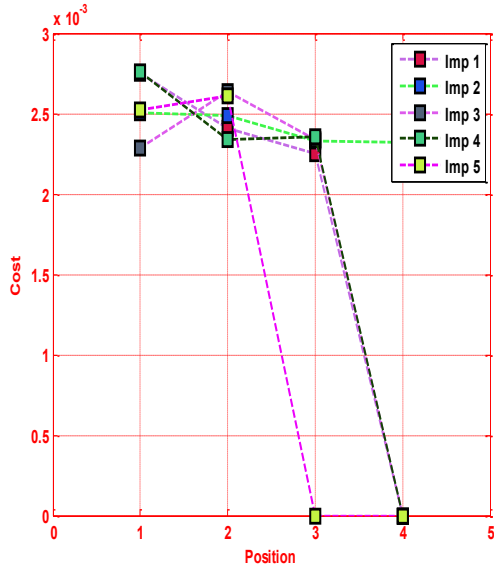


Figure 5.11.a Trend of the input data of the system

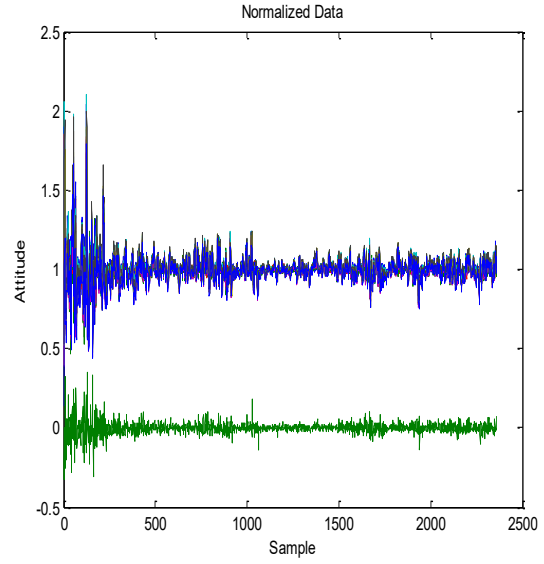


Figure 5.11.b Imperialist competition

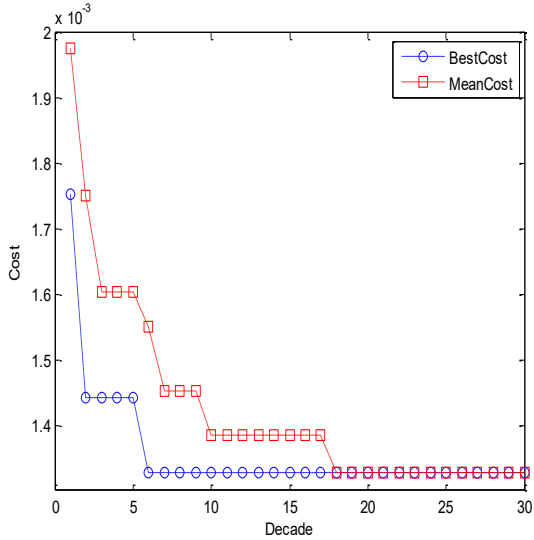


Figure 5.11.c The costs trend in the algorithm stages

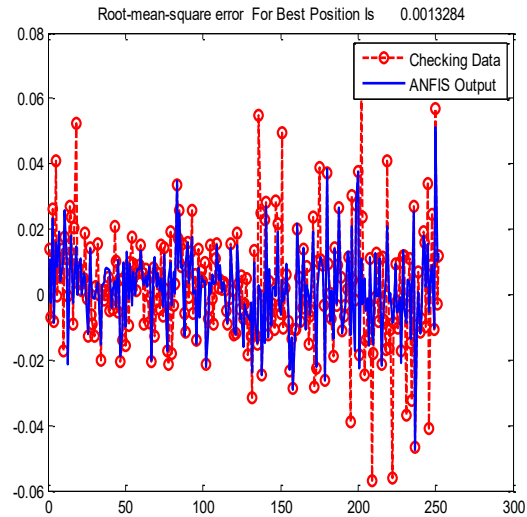


Figure 5.11.c The real and fuzzy output

Lee and Jo (K. Lee & Jo, 1999) believe that if the hit ratio defined by $\frac{\text{Number of successes}}{\text{Total number of signals}}$ is above 51%, the

model is regarded as useful. The hit ratios of the new model for the first and second approaches are 85% and 87% respectively while the second approach gives more number of buy and sell signals.

In the base study of Jasemi et al. (Jasemi, et al., 2011) the hit ratio for the first and second approaches are 75% and 74%. Unlike the base study, our results show the superiority of the second approach.

The percentages of correct signals for 1-day period are 40% and 43% for the first and second approaches respectively. It is to be noted that increasing the learning period's length does not have a sensible effect on the results.

Table 5-4 shows the prediction results for 1-day and 6-day periods and also the total number of buy and sell signals. The correct signals are achieved from comparing the predictions with real happenings of the stock market.

Table 5-4 The correct prediction percentage for 1 and 6 day periods with total number of buy and sell signals

Raw Database								Signal Database							
No	1d (%)	6d (%)	Sig. No	No	1d (%)	6d (%)	Sig. No	No	1d (%)	6d (%)	Sig. No	No	1 d (%)	6d (%)	Sig. No
1	0.33	0.93	175	25	0.35	0.84	146	1	0.27	0.94	177	25	0.33	0.87	124
2	0.32	0.93	147	26	0.35	0.8	148	2	0.33	0.92	123	26	0.28	0.83	168
3	0.36	0.9	205	27	0.29	0.91	171	3	0.55	0.9	168	27	0.34	0.95	122
4	0.35	0.93	205	28	0.34	0.9	191	4	0.4	0.93	161	28	0.55	0.92	184
5	0.21	0.63	166	29	0.35	0.9	194	5	0.28	0.65	167	29	0.4	0.96	164
6	0.33	0.85	132	30	0.3	0.87	191	6	0.44	0.87	124	30	0.28	0.67	186
7	0.33	0.84	141	31	0.4	0.87	130	7	0.34	0.88	163	31	0.37	0.69	183
8	0.33	0.93	147	32	0.32	0.84	184	8	0.24	0.82	182	32	0.3	0.81	207
9	0.29	0.92	167	33	0.31	0.74	180	9	0.55	0.9	145	33	0.23	0.82	187
10	0.34	0.93	164	34	0.32	0.92	184	10	0.61	0.96	165	34	0.58	0.97	175
11	0.36	0.93	192	35	0.37	0.94	173	11	0.43	0.94	150	35	0.57	0.95	211
12	0.33	0.69	168	36	0.2	0.65	131	12	0.21	0.9	154	36	0.43	0.93	174
13	0.32	0.89	145	37	0.37	0.89	142	13	0.41	0.87	143	37	0.51	0.91	211
14	0.35	0.81	156	38	0.37	0.83	139	14	0.34	0.84	149	38	0.36	0.81	170
15	0.33	0.87	151	39	0.35	0.83	145	15	0.27	0.8	184	39	0.29	0.85	181
16	0.29	0.9	185	40	0.37	0.92	186	16	0.61	0.94	123	40	0.36	0.94	176
17	0.34	0.92	186	41	0.18	0.68	158	17	0.54	0.96	171	41	0.49	0.91	143
18	0.16	0.63	124	42	0.43	0.9	150	18	0.2	0.64	166	42	0.4	0.96	172
19	0.4	0.89	144	43	0.37	0.84	131	19	0.29	0.85	142	43	0.35	0.83	164
20	0.69	0.9	155	44	0.39	0.82	140	20	0.28	0.87	177	44	0.25	0.83	184
21	0.42	0.71	149	45	0.18	0.64	107	21	0.21	0.75	181	45	0.42	0.92	136
22	0.33	0.91	199	46	0.4	0.89	149	22	0.35	0.94	137	46	0.36	0.85	154
23	0.18	0.57	152	47	0.38	0.83	136	23	0.33	0.89	168	47	0.37	0.82	154
24	0.38	0.87	133	48	0.35	0.84	146	24	0.44	0.94	133	48	0.26	0.84	172

Figure 5.12 shows the *RMSE* changes for the datasets of 1, 2, 3 and 4 in the first approach for the different number of features from 3 to 15. Figure 5.13 and Figure 5.14 show the *RMSE* changes for the different number of features and datasets. The first approach results show that all of the 15 features are important and have the best prediction; while by increasing the number of features in any of the 48 data sets, *RMSE* is decreased.

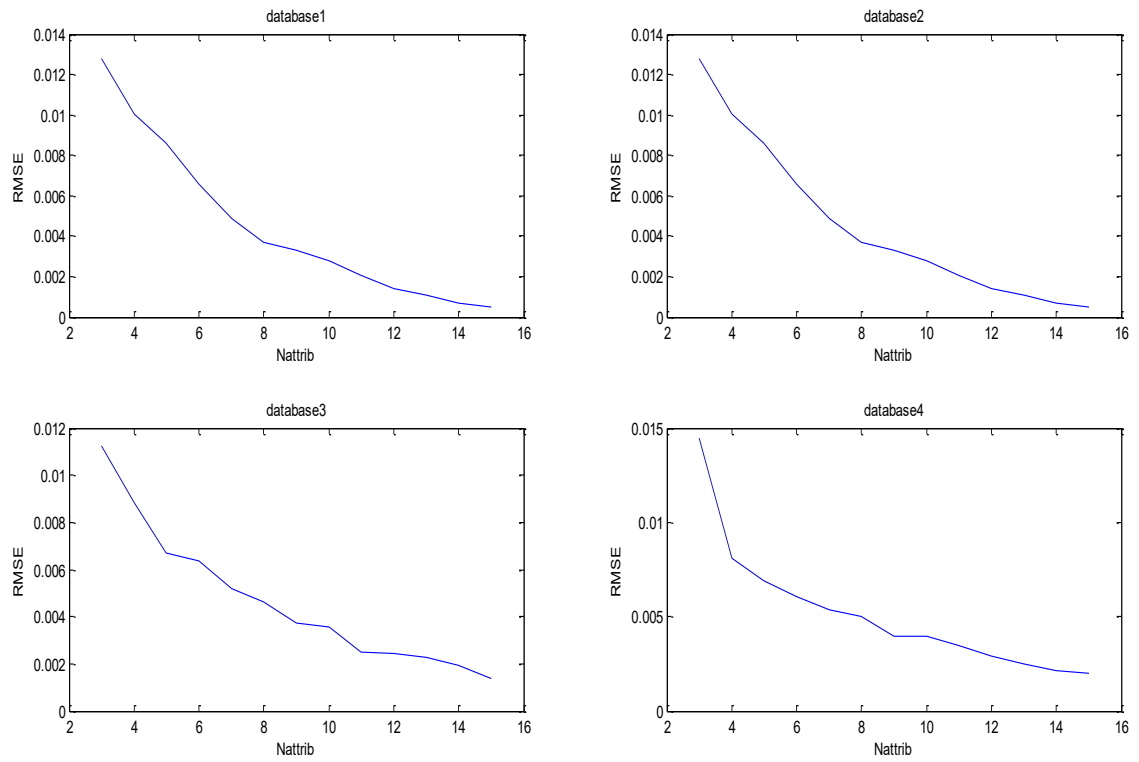


Figure 5.12 *RMSE* changing for the first approach in the first four datasets

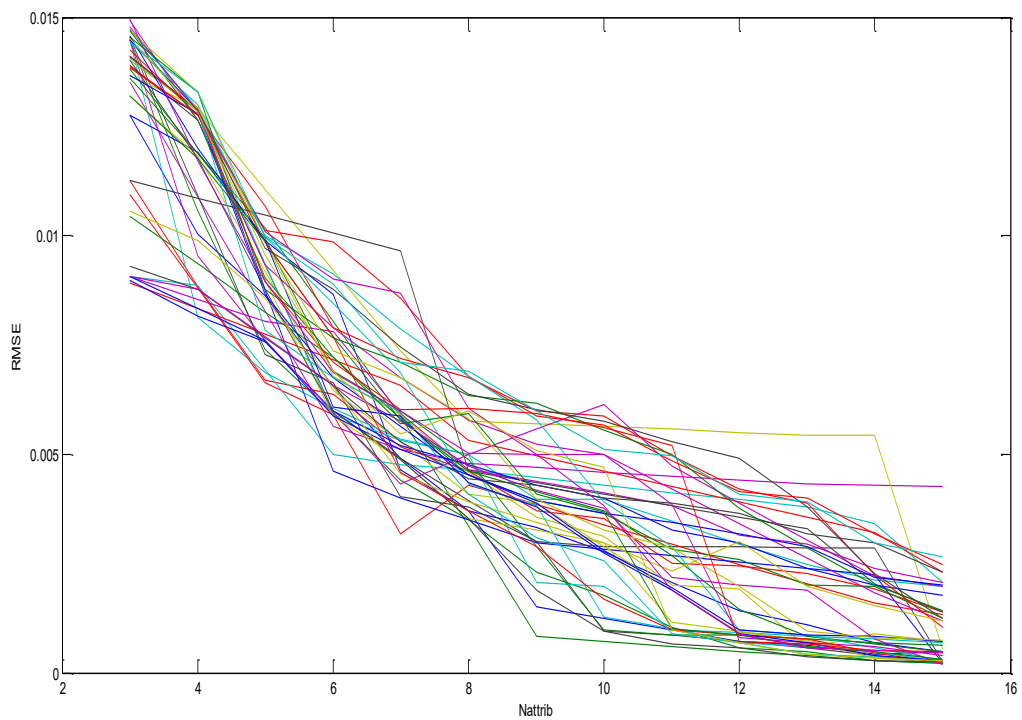


Figure 5.13 *RMSE* changing for the first approach in the 48 datasets

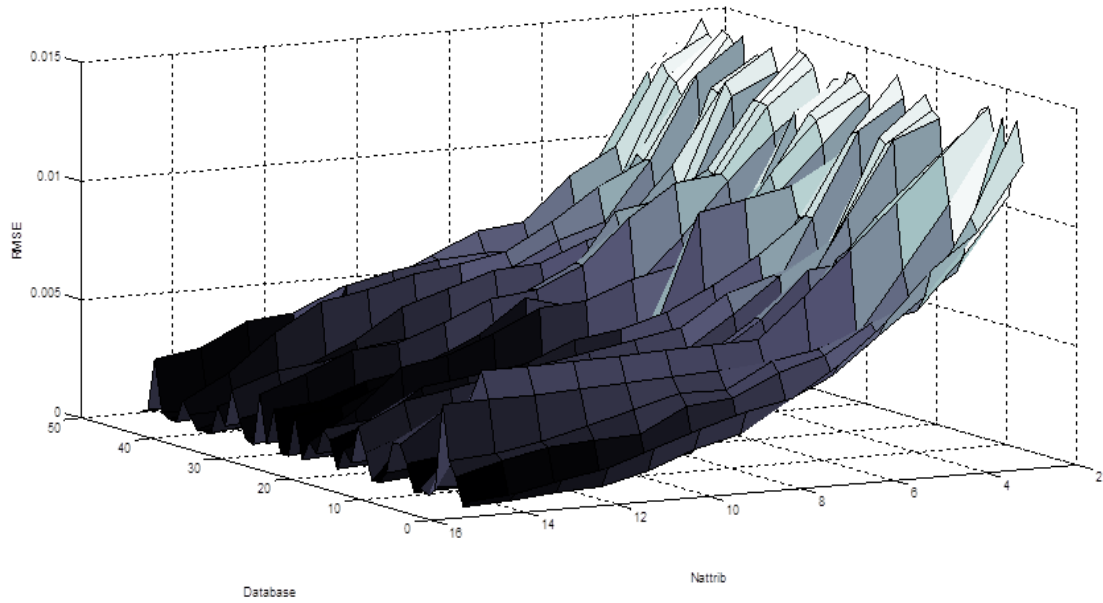


Figure 5.14 *RMSE* changing for the first approach with different number of features (N_{attrib}) for each dataset

Figure 5.15 shows RMSEs in different numbers of features for the first four datasets in the second approach. In datasets 1, 2 and 4 respectively 15, 10 and 20 features have the minimum RMSE. The RMSE changes in all datasets are shown by Figure 5.16. Also, Figure 5.17 shows the RMSE changes with the different number of features for all datasets. Results of this approach show that the feature selection is very important and affect the model efficiency considerably. In most of the second approach data sets, RMSE decreases by increasing the number of features and gets its minimum for a particular number of features and then increases. The best result in almost all of the data sets occurred in an interval from 15 to 20 features.

It should be noted that the main aim of feature selection is to obtain fewer but more effective numbers of features and simultaneously minimizing the loss of the model's prediction accuracy. The results indicated that, by using 15 features, our obtaining accuracy is less than using the second approach with 24 features. However, in the second approach, we find the optimized number of features based on wrapper algorithm. Our results in finding the most effective features can enable the investors to analyze the market with fewer features, and not getting confused in the market by many features which are not necessarily effective.

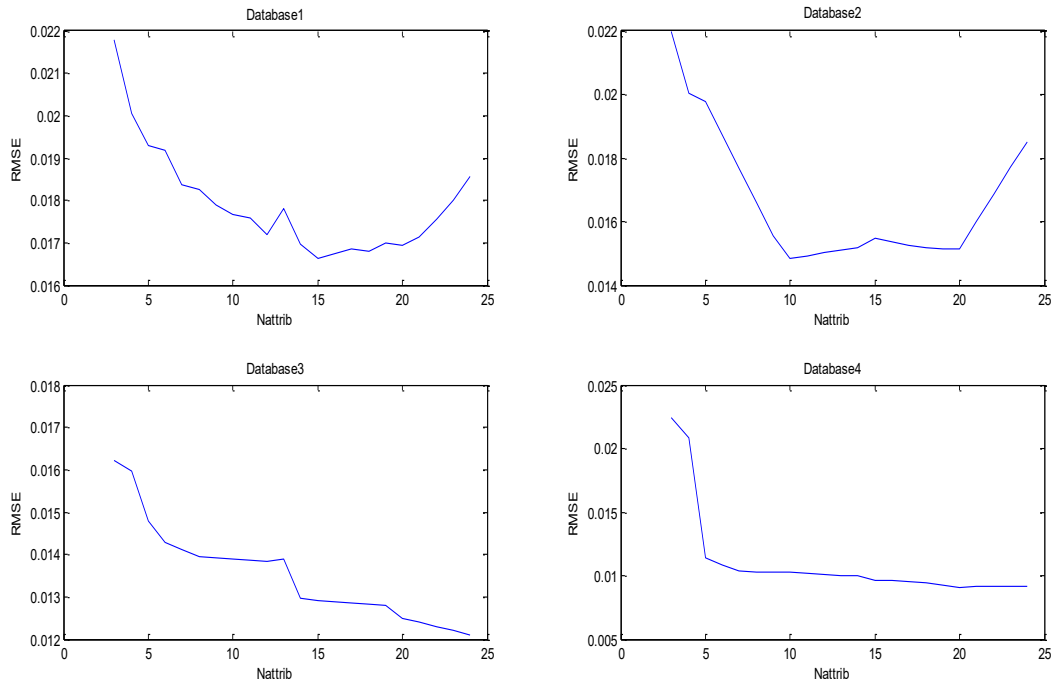


Figure 5.15 RMSE changing for the second approach in the first four datasets

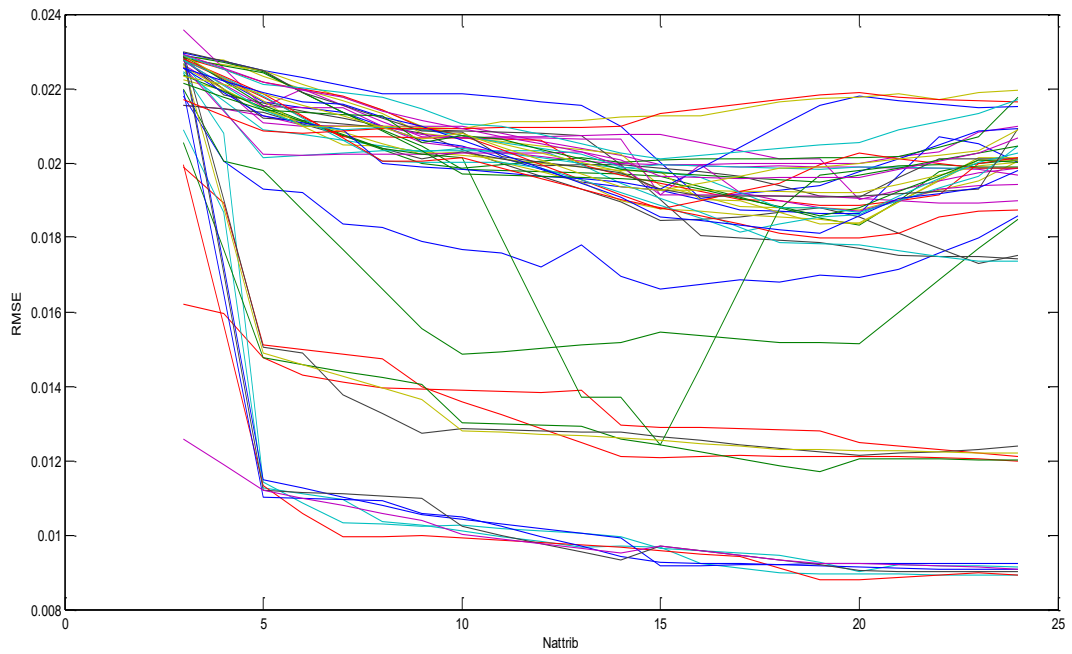


Figure 5.16 *RMSE* changing for the second approach in the 48 datasets

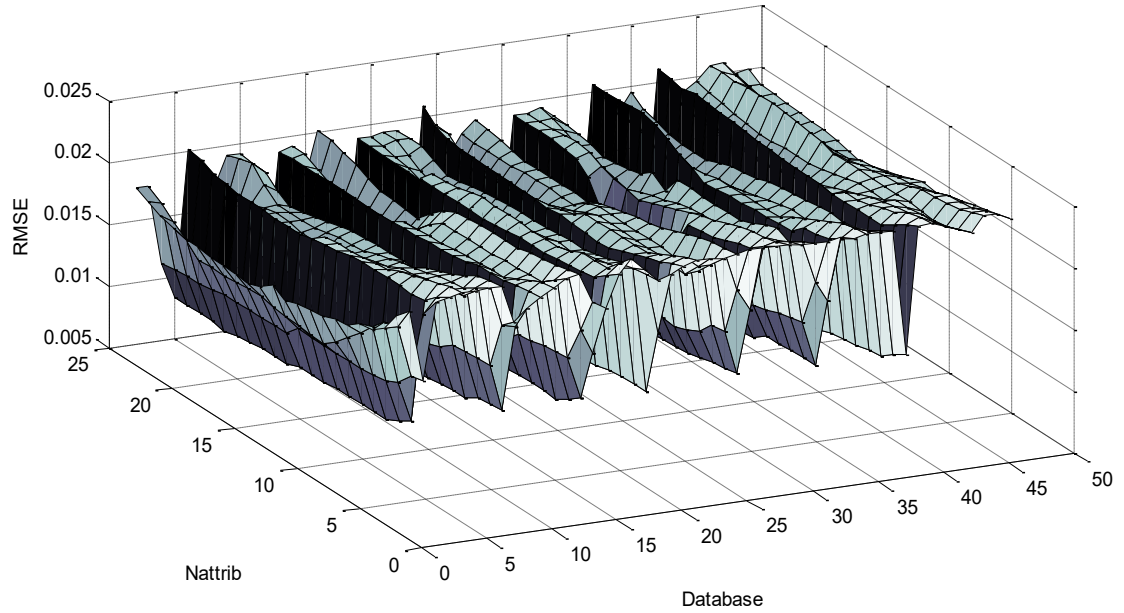


Figure 5.17 RMSE changing for the second approach with different number of features for each dataset

It is to be noted that the features are not the same in different datasets. Table 5-5 shows the optimum features and their best and mean costs for each dataset.

Table 5-5 Optimal features and their costs in second approach

Dataset	n-feature	Best sol-position	Best Cost(RMSE)	Mean Cost(RMSE)
1	15	[2 20 5 11 16 24 1 7 14 3 17 10 23 15 9]	0.016624	0.016624
2	10	[2 13 16 6 7 12 24 19 22 18]	0.0048508	0.0048508
3	24	[14 15 11 20 23 17 6 24 4 9 22 21 12 8 5 16 3 19 18 13 2 10 1 7]	0.012097	0.012097
4	20	[19 21 1 22 7 4 18 3 10 9 14 12 8 2 24 13 17 23 15 11]	0.0090231	0.0090231
5	15	[8 1 20 19 23 5 9 6 22 21 11 13 10 17 24]	0.019963	0.019963
6	20	[16 5 22 8 23 9 18 17 15 6 21 7 3 4 10 1 20 19 24 13]	0.018746	0.018746
7	23	[1 12 20 24 6 13 22 5 3 8 21 15 9 23 7 14 10 2 19 16 4 17 18]	0.017296	0.017296
8	20	[8 23 24 13 10 1 20 9 18 17 15 6 21 7 3 4 19 16 5 22]	0.018647	0.018647
9	20	[12 7 23 21 3 8 19 22 6 17 4 2 15 18 20 16 5 13 10 1]	0.018313	0.018313
10	24	[15 11 18 23 2 20 7 9 1 14 10 13 19 6 22 3 5 12 16 8 17 21 4 24]	0.012	0.012
11	24	[6 20 17 16 18 19 15 7 8 23 3 4 1 22 2 13 21 11 9 12 10 24 14 5]	0.008919	0.008919
12	24	[20 6 17 14 24 16 18 19 21 11 9 15 8 7 23 3 4 1 22 2 12 13 10 5]	0.019668	0.019668
13	20	[13 11 6 24 18 9 1 12 3 19 23 17 4 21 16 5 22 10 8 2]	0.018401	0.018401
14	24	[11 8 12 22 1 16 15 4 10 5 24 20 14 13 17 3 2 23 18 19 6 9 21 7]	0.017431	0.017431
15	20	[7 5 22 8 3 17 15 4 24 13 10 1 20 9 18 19 16 23 6 21]	0.019758	0.019758
16	24	[12 18 17 2 22 6 21 15 8 4 11 1 20 23 14 7 13 9 3 19 10 16 24 5]	0.01201	0.01201
17	20	[10 24 20 2 23 7 17 18 9 15 4 1 22 13 14 11 19 21 8 3]	0.008793	0.008793
18	24	[8 23 1 11 20 4 14 7 12 18 17 2 15 13 22 6 21 9 19 3 24 5 10 16]	0.019862	0.019862
19	20	[13 21 11 5 16 24 6 22 18 9 1 12 3 10 2 19 23 17 4 8]	0.019008	0.019008
20	20	[2 1 16 11 17 12 20 6 10 21 5 18 13 3 23 9 7 8 22 19]	0.018369	0.018369
21	21	[18 19 5 6 3 1 11 10 15 22 4 13 9 17 2 23 7 20 16 8]	0.019075	0.019075
22	24	[6 16 18 23 3 4 1 2 22 13 11 20 17 21 19 15 7 9 8 12 10 14 24 5]	0.009078	0.009078
23	10	[5 6 15 7 3 13 9 17 16 21]	0.01986	0.01986
24	20	[19 23 5 24 6 22 13 21 11 9 1 4 14 7 24 20 2 16 12 19]	0.018852	0.018852
25	23	[18 2 13 19 6 22 3 5 12 16 8 17 21 4 24 20 7 15 11 9 1 14 10]	0.017346	0.017346
26	23	[20 1 21 23 22 10 9 19 7 8 24 11 2 13 12 14 15 17 3 5 4 18 16 6]	0.01891	0.01891
27	20	[4 13 19 18 5 3 6 1 11 10 15 22 9 17 2 7 23 20 8 16]	0.019189	0.019189
28	20	[8 24 13 21 15 11 7 19 22 14 4 12 20 5 1 6 3 2 9 16]	0.012154	0.012154
29	15	[3 22 11 20 23 18 1 10 2 12 4 7 5 6 17]	0.0091685	0.0091685
30	20	[14 24 4 12 10 13 8 9 1 17 21 15 3 16 23 22 2 6 20 7]	0.020243	0.02043
31	17	[5 4 19 6 21 10 20 7 22 11 13 16 18 23 8 1 9]	0.018689	0.018689
32	20	[9 8 13 11 7 14 19 16 4 5 22 24 12 21 6 3 18 17 2 15]	0.018616	0.018616
33	20	[15 19 5 3 7 1 13 11 8 14 20 21 22 9 6 16 17 18 12 10]	0.019793	0.019793
34	24	[19 23 3 15 17 11 24 4 21 22 14 2 13 5 10 6 9 8 20 18 7 12 16 1]	0.012201	0.012201

Dataset	n-feature	Best sol-position	Best Cost(RMSE)	Mean Cost(RMSE)
35	24	[14 12 9 20 7 2 3 22 16 1 21 8 18 19 11 24 10 17 4 5 15 23 13 6]	0.0090162	0.0090162
36	14	[11 22 21 24 4 20 8 17 7 16 23 5 1 10]	0.019299	0.019299
37	20	[24 20 15 12 2 8 6 11 18 22 4 9 5 10 7 16 3 1 17 21]	0.018792	0.018792
38	20	[9 15 17 18 24 5 21 22 1 19 12 20 2 3 10 14 23 16 7 13]	0.017977	0.017977
39	20	[9 5 21 22 1 20 2 3 10 19 12 15 17 18 24 16 7 14 23 13]	0.018675	0.018675
40	24	[23 22 10 20 1 13 12 14 15 17 3 4 5 21 9 19 7 8 24 11 2 18 6 16]	0.009077	0.009077
41	10	[11 7 14 19 16 12 20 24 13 21]	0.020924	0.020924
42	17	[16 1 20 7 8 17 2 10 14 19 23 5 15 4 22 9 6]	0.019094	0.019094
43	15	[18 19 2 17 13 1 22 4 15 9 24 23 6 10 3]	0.018561	0.018561
44	20	[15 17 9 5 12 18 24 22 1 20 16 21 2 3 10 19 7 23 13 14]	0.019678	0.019678
45	4	[1 17 13 15 4 22]	0.020789	0.020789
46	15	[4 3 6 17 22 23 16 18 1 5 20 12 9 11 8]	0.020094	0.020094
47	20	[16 8 11 12 2 13 22 4 5 15 20 6 17 1 18 19 24 21 9 7]	0.019602	0.019602

The wrapper ANFIS-ICA output of this study shows that feature selection by ICA has a good impact on the accuracy of ANFIS for different datasets. We also compare our method with Jasemi et al (Jasemi, et al., 2011) which uses a feed-forward neural network (NN) optimized by a back propagate algorithm and ANFIS-GA wrapper model which used genetic algorithms (GA) instead of ICA for feature selection. The results accuracy shows that wrapper ICA-ANFIS algorithm dominates the NN and also wrapper ANFIS-GA algorithm. In more detail discussion, the ANFIS model has solely better results than NN algorithm. Forecasting accuracy of NN depends on learning data set and their adequacy. Moreover, NN methods sometimes get stuck in a local minimum, so choosing proper dataset is too critical in neural network models. In the other side, the ANFIS has better result when it deals with the noisy and inadequate data set. Based on the fact that in stock exchange dataset we always encounter with noisy data, enjoying from ANFIS model is preferable than NN models. Also, the ICA can diversify the dataset for learning issues better than other models such as the GA model.

Table 5-6 Comparison between algorithms

	Total Hit ratio	Hit ratio Approach 1	Hit ratio Approach 2	Hit ratio 1-day/ Approach 1	Hit ratio 1-day/ Approach 2	Mean Runtime (s)	Mean Cost (RMSE)
NN	74.2 %	74.8 %	73.6 %	45 %	43.4 %	20	0.0166
ANFIS-GA	77 %	76 %	78 %	35 %	38 %	110.9	0.0062
ANFIS-ICA	86 %	85 %	87 %	40 %	43 %	34.2	0.0048

It is considerable to mention that ICA algorithm gets the final response more quickly than GA and because wrapper models are time-consuming and need a significant amount of time to convergence, this algorithm proves its ability in another point of view.

5.5.Conclusion

The fifth chapter proposes a novel methodology for feature selection and time series estimation based on a wrapper approach called ANFIS-ICA, which results from the combination of ANFIS, a fuzzy logic based

function estimator that adopts the adaptive systems framework to facilitate learning and adaptation, and ICA, an evolutionary algorithm used for subset evaluation. Additionally, the proposed model is applied based on the technical analysis of Japanese Candlestick by two approaches of Signal and Raw database. The chapter incorporates the technique of Japanese Candlestick charts for data transformation and attribute generation. Finally, daily stock prices for the General Motors Company at the New York Stock Exchange from 2000 to 2009 are applied for benchmarking.

The root of mean square errors, the number, and percentage of the signals that are predicted correctly are considered as the assessment measures while the results show a good reliability of the model that is even better than the base study and model of Jasemi et al. (Jasemi, et al., 2011).

In the first approach, all the 15 features together have the best approximation but in the second approach, feature selection shows its effect on the results very well, while in different datasets almost 20 numbers of features are more appropriate. As a matter of fact it is believed that the analysis of different features or different subsets of them has a significant influence on prediction accuracy. The second approach has a better performance than the first one. Also, it should be noted that the prediction accuracy of this model is better than some other time series models and shows great abilities for decision making about stock trading. Future research directions of the chapter include but are not limited to

1. Combining prediction methods in the framework of fusion models or optimize the classification algorithms by applying some meta-heuristics algorithms to improve the prediction results
2. Predicting the other important variable (in addition to return) such as liquidity (Barak, et al., 2013).
3. Using fundamental features and textual information, in addition to technical features, in order to have a more comprehensive features and to be able to predict long term situation of stocks
4. Prediction of stocks in other popular stock markets such as DAX or NASDAQ
5. Applying statistical feature selection models such as Filter methods and CFS (X. Zhang, Hu, Xie, Wang, et al., 2014), or using other wrapper models to compare the results

Chapter 6

Conclusion and Future Research

Machine learning has become more accessible in many industries within the last few years which especially has transformed the finance field. There are many application of the implementing machine learning models to manage a portfolio of financial instruments with the aim of increasing the wealth. However, most of financial institutions do not openly reveal their AI approaches to manage their portfolio, but it is believed that machine learning and deep learning methods are playing as an important role in their portfolio management decisions.

In order to provide an accurate prediction, using suitable features which can explain better target values' variations is essential. Thus, the crucial section of machine learning is feature selection. With an appropriate feature selection, the prediction model not only will be more accurate, but also be potent and reliable. Besides, efficient time consumption in model training can be another beneficiary of the good feature selection.

Two most important variables which are desirable to be forecasted accurately in financial markets, are risk and return of financial instruments. For the forecasting instrument prices, there two common approaches including fundamental and technical evaluation. In the fundamental analysis, investors evaluate intrinsic value of stocks which came from its future cash flow, while technical analysts implement some technical tools to predict price movements. So, here, with a complete view to the stock price forecasting criteria, a review of three researches of author that consider a diverse and appropriate pool of features which came from both fundamental and technical analysis, are provided.

In chapter three, the author has developed a novel approach to predict stocks' return and risk using a three-stage method through which, risk and return are predicted by applying data mining techniques for the given fundamental features. In this chapter, a hybrid algorithm based on the filter and function-based clustering is developed. The results show that the proposed hybrid model is a proper tool for effective feature selection and these features are good indicators for the prediction of risk and return. To illustrate the approach as well as to train data and test, Tehran Stock Exchange (TSE) data from 2002 to 2011 was applied.

After providing a robust method for feature selection, as it's mentioned that is a critical section of implementing machine learning models, in chapter four, the author provides a method to create a potent predictor. In chapter four, a fusion model based on the use of multiple diverse base classifiers is proposed that operate on a common input and a meta-classifier that learns from base classifiers' outputs to obtain a

more precise stock return and risk predictions. A set of diversity methods, including Bagging, Boosting, and AdaBoost is applied to create diversity in classifier combinations. Moreover, the number and procedure for selecting base classifiers for fusion schemes are determined using a methodology based on dataset clustering and candidate classifiers' accuracy. The results demonstrate that Bagging exhibited superior performance within the fusion scheme and could achieve a maximum of 83.6% accuracy with stacking of Decision Tree, LAD Tree and Rep Tree for return prediction and 88.2% accuracy with stacking of BF Tree, DTNB and LAD Tree in risk prediction. For feature selection part, a wrapper-GA algorithm is developed and compared with the fusion model. This chapter seeks to help researcher selecting the best individual classifiers and fusing the proper scheme in stock market prediction. Again, the model is applied to the Tehran Stock Exchange (TSE) data for the period from 2002 to 2012.

In addition to using fundamental features, chapter five presents a novel forecasting model for stock markets based on the wrapper ANFIS (Adaptive Neural Fuzzy Inference System) – ICA (Imperialist Competitive Algorithm) and technical analysis of Japanese Candlestick. The Author has implemented two approaches of Raw-based and Signal-based to extract the model's input variables with 15 and 24 features, respectively. The correct predictions percentages for periods of 1- 6 days with the total number of buy and sell signals are considered as output variables. In the proposed model, the ANFIS prediction results are used as a cost function of the wrapper model, and ICA is used to select the most appropriate features. This novel combination of feature selection not only takes advantage of ICA optimization swiftness but also the ANFIS prediction accuracy. The emitted buy and sell signals of the model revealed that Signal databases approach gets better results with 87% prediction accuracy and the wrapper features selection obtains 12% improvement in predictive performance regarding the base study. Additionally, since the wrapper-based feature selection models are considerably more time-consuming, the presented wrapper ANFIS-ICA algorithm's results have superiority in time decreasing as well as prediction accuracy increasing regarding other algorithms such as wrapper Genetic algorithm (GA).

In these three chapters, new machine learning methods are implemented to create a more accurate predictor; especially, providing the novel feature selections with considering both fundamental and technical features to forecast stock market prices. Additionally, the author considered both sides of long-term and short-term of investment viewpoint with providing different forecasting horizon (quarterly, using fundamental features in chapters three and four, and, daily, using technical features in chapter five).

There are some limitation among the presented chapters which could be improved in the future researches. Some limitations are related to the data gathering part, such as fundamental data in Chapter 3 and 4, which were the financial reports of companies in Tehran stock exchange. Because the reports were available quarterly, therefore the forecasting horizon was set for one step ahead quarter. Thus, short-term trading does not consider in those two chapters; however, the results for Chapter 5 is pertinent for the short-term traders.

From the methodology aspect, it is noteworthy to point to the Fusion model disadvantage. In this model, using multiple steps to combine a base model with diversity and fusion methods, could consume more time to find the best combination. This could be considered as a weakness in compares with new and well-known models like Random Forest or XGBoost models. Moreover, in this thesis, the risk and return are predicted individually (chapter three and four), but for the future works, we can combine the risk and return linearly and then cluster their result into some groups as new labels. Then the classification task will be the prediction of the class of risk and return simultaneously.

Further researches in directions of these topics include but are not limited to:

Optimizing the parameters of classification algorithms using meta-heuristics algorithms to improve the prediction results; Using other decision tree models like Random Forest and XGBoost for classification, Predicting other important response variables (in addition to risk and return) such as liquidity (Barak, et al., 2013); Using technical features and textual information, in addition to fundamentals features, in order to use more comprehensive features and be able to predict the short term situations of stocks; and Customizing the proposed approach for the prediction of risk and return in a particular industry or investigating the accuracy of the procedure using data from other popular stock markets, such as the US stock market, which may result in new dimensions for this procedure.

Literature

- Alceu S. Britto, J., Sabourin, R., & Oliveira, L. E. S. (2014). Dynamic selection of classifiers-A comprehensive review. *Pattern Recogn.*, 47, 3665-3680.
- Araújo, R. d. A., & Ferreira, T. A. (2013). A morphological-rank-linear evolutionary method for stock market prediction. *Information Sciences*, 237, 3-17.
- Asadi, S., Hadavandi, E., Mehmanpazir, F., & Nakhostin, M. M. (2012). Hybridization of evolutionary Levenberg–Marquardt neural networks and data pre-processing for stock market prediction. *Knowledge-Based Systems*, 35, 245-258.
- Atashpaz-Gargari, E., & Lucas, C. (2007). Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on* (pp. 4661-4667): IEEE.
- Atsalakis, G. S., Dimitrakakis, E. M., & Zopounidis, C. D. (2011). Elliott Wave Theory and neuro-fuzzy systems, in stock market prediction: The WASP system. *Expert Systems with Applications*, 38, 9196-9206.
- Atsalakis, G. S., & Valavanis, K. P. (2009). Surveying stock market forecasting techniques–Part II: Soft computing methods. *Expert Systems with Applications*, 36, 5932-5941.
- Bagheri, A., Mohammadi Peyhani, H., & Akbari, M. (2014). Financial forecasting using ANFIS networks with Quantum-behaved Particle Swarm Optimization. *Expert Systems with Applications*, 41, 6235-6250.
- Barak, S., Abessi, M., & Modarres, M. (2013). Fuzzy turnover rate chance constraints portfolio model. *European journal of operational research*, 228, 141-147.
- Barak, S., Dahooie, J. H., & Tichý, T. (2015). Wrapper ANFIS-ICA method to do stock market timing and feature selection on the basis of Japanese Candlestick. *Expert Systems with Applications*, 42, 9221-9235.
- Barak, S., & Modarres, M. (2015). Developing an approach to evaluate stocks by forecasting effective features with data mining methods. *Expert Systems with Applications*, 42, 1325-1339.
- Barak, S., & Sadegh, S. S. (2016). Forecasting energy consumption using ensemble ARIMA–ANFIS hybrid algorithm. *International Journal of Electrical Power & Energy Systems*, 82, 92-104.
- Bartholdy, J., & Peare, P. (2005). Estimation of expected return: CAPM vs. Fama and French. *International Review of Financial Analysis*, 14, 407-427.
- Bauer, R., Guenster, N., & Otten, R. (2004). Empirical evidence on corporate governance in Europe: The effect on stock returns, firm value and performance. *Journal of Asset Management*, 5, 91–104.
- Bernstein, L., & Wild, J. (1999). *Analysis of Financial Statements* (5 edition ed.): McGraw-Hill; .
- Beyaz, E., Tekiner, F., Zeng, X.-j., & Keane, J. (2018). Comparing Technical and Fundamental indicators in stock price forecasting. In *2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)* (pp. 1607-1613): IEEE.
- Bezdek, J. C. (1981). *Pattern recognition with fuzzy objective function algorithms*: Kluwer Academic Publishers.
- Biabangard-Oskouyi, A., Atashpaz-Gargari, E., Soltani, N., & Lucas, C. (2008). Application of imperialist competitive algorithm for materials property characterization from sharp indentation test. *To be appeared in the International Journal of Engineering Simulation*.
- Bisoi, R., & Dash, P. K. (2014). A hybrid evolutionary dynamic neural network for stock market trend analysis and prediction using unscented Kalman filter. *Applied Soft Computing*, 19, 41-56.
- Brealey, R. A., Myers, S. C., & Allen, F. (2007). *Principles of Corporate Finance* (9 ed.): McGraw-Hill.
- Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24, 123-140.
- Cai, J., Luo, J., Wang, S., & Yang, S. (2018). Feature selection in machine learning: A new perspective. *Neurocomputing*, 300, 70-79.
- Cao, J., Kwong, S., & Wang, R. (2012). A noise-detection based AdaBoost algorithm for mislabeled data. *Pattern Recognition*, 45, 4451-4465.
- Cao, Q., Leggio, K. B., & Schniederjans, M. J. (2005). A comparison between Fama and French's model and artificial neural networks in predicting the Chinese stock market. *Computers & Operations Research*, 32, 2499-2512.
- Carnes, T. A., & College, B. (2006). Unexpected changes in quarterly financial-statement line items and their relationship to stock prices. *Academy of Accounting and Financial Studies Journal*, 10, 99-116.

- Cervelló-Royo, R., Guijarro, F., & Michniuk, K. (2015). Stock market trading rule based on pattern recognition and technical analysis: Forecasting the DJIA index with intraday data. *Expert Systems with Applications*, 42, 5963-5975.
- Chang, P.-C., & Liu, C.-H. (2008). A TSK type fuzzy rule based system for stock price prediction. *Expert Systems with Applications*, 34, 135-144.
- Chang, T.-S. (2011). A comparative study of artificial neural networks, and decision trees for digital game content stocks price prediction. *Expert Systems with Applications*, 38, 14846-14851.
- Chavarnakul, T., & Enke, D. (2008). Intelligent technical analysis based equivolume charting for stock trading using neural networks. *Expert Systems with Applications*, 34, 1004-1017.
- Chen, Y.-S., & Cheng, C.-H. (2012). A soft-computing based rough sets classifier for classifying IPO returns in the financial markets. *Applied Soft Computing*, 12, 462-475.
- Chen, Y., Mabu, S., Shimada, K., & Hirasawa, K. (2009). A genetic network programming with learning approach for enhanced stock trading model. *Expert Systems with Applications*, 36, 12537-12546.
- Cheng, J.-H., Chen, H.-P., & Lin, Y.-M. (2010a). A hybrid forecast marketing timing model based on probabilistic neural network, rough set and C4.5. *Expert Systems with Applications*, 37, 1814-1820.
- Cheng, J.-H., Chen, H.-P., & Lin, Y.-M. (2010b). A hybrid forecast marketing timing model based on probabilistic neural network, rough set and C4. 5. *Expert Systems with Applications*, 37, 1814-1820.
- Cheng, S.-H. (2014). A Hybrid Predicting Stock Return Model Based on Bayesian Network and Decision Tree. In M. Ali, J.-S. Pan, S.-M. Chen & M.-F. Horng (Eds.), *Modern Advances in Applied Intelligence* (Vol. 8482, pp. 218-227): Springer International Publishing.
- Cheng, S.-H. (2015). A Hybrid Predicting Stock Return Model Based on Logistic Stepwise Regression and CART Algorithm. In N. T. Nguyen, B. Trawiński & R. Kosala (Eds.), *Intelligent Information and Database Systems* (Vol. 9011, pp. 624-633): Springer International Publishing.
- Chiang, W.-C., Urban, T. L., & Baldridge, G. W. (1996). A neural network approach to mutual fund net asset value forecasting. *Omega*, 24, 205-215.
- Covões, T. F., & Hruschka, E. R. (2011). Towards improving cluster-based feature selection with a simplified silhouette filter. *Information Sciences*, 181, 3766-3782.
- Crone, S. F., & Kourentzes, N. (2010). Feature selection for time series prediction—A combined filter and wrapper approach for neural networks. *Neurocomputing*, 73, 1923-1936.
- Dasarathy, B. V., & Sheela, B. V. (1979). A composite classifier system design: Concepts and methodology. *Proceedings of the IEEE*, 67, 708-713.
- Dastgir, M., & Afshari, M. H. (2004). Evaluating Stock Pricing models on Tehran Stock Exchange Assets. *Accounting Studies*, 3, 60-94.
- de Oliveira, D. F., Canuto, A. M. P., & de Souto, M. C. P. (2009). Use of multi-objective genetic algorithms to investigate the diversity/accuracy dilemma in heterogeneous ensembles. In *Neural Networks, 2009. IJCNN 2009. International Joint Conference on* (pp. 2339-2346).
- de Oliveira, F. A., Nobre, C. N., & Zárate, L. E. (2013). Applying Artificial Neural Networks to prediction of stock price and improvement of the directional prediction index – Case study of PETR4, Petrobras, Brazil. *Expert Systems with Applications*, 40, 7596-7606.
- do Prado, H. A., Ferneda, E., Morais, L. C., Luiz, A. J., & Matsura, E. (2013). On the Effectiveness of Candlestick Chart Analysis for the Brazilian Stock Market. *Procedia Computer Science*, 22, 1136-1145.
- duda, R. o., Hart, P. E., & Strok, D. G. (2001). *Pattern Classification*: Wiley.
- Dumais, S., Platt, J., Heckerman, D., & Sahami, M. (1998). Inductive learning algorithms and representations for text categorization. In *In Proceedings of the International Conference on Information Knowledge Management* (pp. 148–155).
- Engel, J. (1988). Polytomous logistic regression. *Statistica Neerlandica*, 42, 233-252.
- Enke, D., & Thawornwong, S. (2005). The use of data mining and neural networks for forecasting stock market returns. *Expert Systems with Applications*, 29, 927-940.
- Esfahanipour, A., & Aghamiri, W. (2010). Adapted neuro-fuzzy inference system on indirect approach TSK fuzzy rule base for stock market analysis. *Expert Systems with Applications*, 37, 4742-4748.
- Esposito, F., Malerba, D., Semeraro, G., & Kay, J. (1997). A comparative analysis of methods for pruning decision trees. *IEEE transactions on pattern analysis and machine intelligence*, 19, 476-491.
- Fama, E. F., & French, K. R. (1993). Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics*, 33, 3-56.
- Fama, E. F., & French, K. R. (2012). Size, value, and momentum in international stock returns. *Journal of Financial Economics*, 105, 457-472.

- Ferreiro, S., Sierra, B., Irigoien, I., & Gorritxategi, E. (2011). Data mining for quality control: Burr detection in the drilling process. *Computers & Industrial Engineering*, 60, 801-810.
- Finlay, S. (2011). Multiple classifier architectures and their application to credit risk assessment. *European journal of operational research*, 210, 368-378.
- Frank, E., & Witten, I. H. (1998). Generating Accurate Rule Sets Without Global Optimization. In *Fifteenth International Conference on Machine Learning*, (pp. 144-151).
- Freund, Y. (1995). Boosting a Weak Learning Algorithm by Majority. *Information and Computation*, 121, 256-285.
- Freund, Y., & Schapire, R. E. (1996). Experiments with a New Boosting Algorithm. *Machine Learning: Proceedings of the Thirteenth International Conference*, 48-56.
- Freund, Y., & Schapire, R. E. (1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, 55, 119-139.
- Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning* (Vol. 1): Springer series in statistics New York.
- Fumera, G., & Roli, F. (2005). A theoretical and experimental analysis of linear combiners for multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27, 942-956.
- Gargari, E. A., Hashemzadeh, F., Rajabioun, R., & Lucas, C. (2008). Colonial competitive algorithm: a novel approach for PID controller design in MIMO distillation column process. *International Journal of Intelligent Computing and Cybernetics*, 1, 337-355.
- Gershenfeld, N. A. (1999). *The nature of mathematical modeling*: Cambridge university press.
- Giraud-Carrier, C. (2008). Metalearning-a tutorial. In *Proceedings of the 7th international conference on machine learning and applications*.
- Gordon, M. J. (1982). *The investment, financing, and valuation of the corporation* (Vol. 52): Greenwood Press Reprint.
- Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, 3, 1157-1182.
- Hafezi, R., Shahrabi, J., & Hadavandi, E. (2015). A bat-neural network multi-agent system (BNNMAS) for stock price prediction: Case study of DAX stock price. *Applied Soft Computing*, 29, 196-210.
- Haghighi, M. S., Vahedian, A., & Yazdi, H. S. (2011). Creating and measuring diversity in multiple classifier systems using support vector data description. *Applied Soft Computing*, 11, 4931-4942.
- Hall, M. (1998). *Correlation-based feature subset selection for machine learning*. University of Waikato.
- Hall, M., & Frank, E. (2008). Combining Naïve Bayes and Decision Tables. In *In Proceedings of the 21st Florida Artificial Intelligence Society Conference (FLAIRS)*. Florida.
- Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11, 10-18.
- Han, J., & Kamber, M. (2006). *Data Mining: Concepts and Techniques*: The Morgan Kaufmann.
- Hernández-Lobato, J. M., & Adams, R. (2015). Probabilistic backpropagation for scalable learning of bayesian neural networks. In *International Conference on Machine Learning* (pp. 1861-1869).
- Hjalmarsson, E. (2010). Predicting global stock returns. *Journal of Financial and Quantitative Analysis*, 45, 49-80.
- Ho, T., Hull, J. J., & Srihari, S. (1994). Decision combination in multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16, 66-75.
- Hofmann, M., & Klinkenberg, R. (2013). *RapidMiner: Data Mining Use Cases and Business Analytics Applications*: Chapman and Hall/CRC.
- Holte, R. C. (1993). Very Simple Classification Rules Perform Well on Most Commonly Used Datasets. *Machine Learning*, 11, 63-90.
- Hong, T.-P., & Wu, C.-W. (2011). Mining rules from an incomplete dataset with a high missing rate. *Expert Systems with Applications*, 38, 3931-3936.
- Huang, C.-F. (2012). A hybrid stock selection model using genetic algorithms and support vector regression. *Applied Soft Computing*, 12, 807-818.
- Huang, C.-J., Yang, D.-X., & Chuang, Y.-T. (2008). Application of wrapper approach and composite classifier to the stock trend prediction. *Expert Systems with Applications*, 34, 2870-2878.
- Huang, C.-L., & Tsai, C.-Y. (2009). A hybrid SOFM-SVR with a filter-based feature selection for stock market forecasting. *Expert Systems with Applications*, 36, 1529-1539.
- Huang, C. F. (2012). A hybrid stock selection model using genetic algorithms and support vector regression. *Applied Soft Computing* 12, 807-818.
- Huang, J., Cai, Y., & Xu, X. (2007). A hybrid genetic algorithm for feature selection wrapper based on mutual information. *Pattern Recognition Letters*, 28, 1825-1844.

- Jang, J.-S. (1993). ANFIS: adaptive-network-based fuzzy inference system. *Systems, Man and Cybernetics, IEEE Transactions on*, 23, 665-685.
- Jasemi, M., Kimiagari, A. M., & Memariani, A. (2011). A modern neural network model to do stock market timing on the basis of the ancient investment technique of Japanese Candlestick. *Expert Systems with Applications*, 38, 3884-3890.
- Kabir, M., & Islam, M. (2010). A new wrapper feature selection approach using neural network. *Neurocomputing*, 73, 3273-3283.
- Kantardzic, M. (2003). Data Mining: Concepts, Models, Methods, and Algorithms. *Technometrics*, 45, 277.
- Kao, L.-J., Chiu, C.-C., Lu, C.-J., & Chang, C.-H. (2013). A hybrid approach by integrating wavelet-based feature extraction with MARS and SVR for stock index forecasting. *Decision Support Systems*, 54, 1228-1244.
- Kaplan, S. N., & Ruback, R. S. (1995). The Valuation of Cash Flow Forecasts: An Empirical Analysis. *The Journal of Finance*, 50, 1059-1093.
- Kar, S., Das, S., & Ghosh, P. K. (2014). Applications of neuro fuzzy systems: A brief review and future outline. *Applied Soft Computing*, 15, 243-259.
- Khan, S. S., & Ahmad, A. (2004). Cluster center initialization algorithm for K-means clustering. *Pattern recognition letters*, 25, 1293-1302.
- Khoshnevisan, B., Bolandnazar, E., Barak, S., Shamshirband, S., Maghsoudlou, H., Altameem, T. A., & Gani, A. (2015). A clustering model based on an evolutionary algorithm for better energy use in crop production. *Stochastic Environmental Research and Risk Assessment*.
- Knorr, E. M., & Ng, R. T. (1999). Finding intentional knowledge of distance-based outliers. In *in Proceedings of the 25th international conference on very large data bases* (pp. 211-222). Edinburgh, Scotland.
- Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, 97, 273-324.
- Kononenko, I. (1994). Estimating attributes: analysis and extensions of relief. In *In Proceedings of the Seventh European Conference on Machine Learning* (pp. 171-182).
- Kourentzes, N., Barrow, D. K., & Crone, S. F. (2014). Neural network ensemble operators for time series forecasting. *Expert Systems with Applications*, 41, 4235-4244.
- Krogh, A., & Vedelsby, J. (1995). Neural network ensembles, cross validation, and active learning. *Advances in Neural Information Processing Systems*, 7, 231-238.
- Kumar, R. (2013). Decision tree for the weather forecasting. *International Journal of Computer Applications*, 76.
- Kuncheva, L. (2004). Combining Pattern Classifiers: Methods and Algorithms. *Wiley-Interscience*.
- Kuncheva, L., & Whitaker, C. (2003). Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy. *Machine Learning*, 51, 181-207.
- Kuncheva, L., Whitaker, C., Shipp, C., & Duin, R. (2003). Limits on the majority vote accuracy in classifier fusion. *Pattern Analysis and Applications*, 6, 22-31.
- Kuncheva, L. I. (2004). *Combining Pattern Classifiers: Methods and Algorithms*: Wiley-Interscience.
- Kwiatkowski, D., Phillips, P. C., Schmidt, P., & Shin, Y. (1992). Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root? *Journal of econometrics*, 54, 159-178.
- Lahmiri, S. (2014). Ensemble with radial basis function neural networks for Casablanca stock market returns prediction. In *Complex Systems (WCCS), 2014 Second World Conference on* (pp. 469-474).
- Lai, R. K., Fan, C.-Y., Huang, W.-H., & Chang, P.-C. (2009). Evolving and clustering fuzzy decision tree for financial time series data forecasting. *Expert Systems with Applications*, 36, 3761-3773.
- Lawrence, S., Giles, C. L., & Tsoi, A. C. (1997). Lessons in neural network training: Overfitting may be harder than expected. In *AAAI/IAAI* (pp. 540-545): Citeseer.
- Lee, K., & Jo, G. (1999). Expert system for predicting stock market timing using a candlestick chart. *Expert Systems with Applications*, 16, 357-364.
- Lee, W.-S., Tzeng, G.-H., Guan, J.-L., Chien, K.-T., & Huang, J.-M. (2009). Combined MCDM techniques for exploring stock selection based on Gordon model. *Expert Systems with Applications*, 36, 6421-6430.
- Levin, N., & Zahavi, J. (2001). Predictive modeling using segmentation. *Journal of Interactive Marketing*, 15, 2-22.
- Lewellen, J. (2004). Predicting returns with financial ratios. *Journal of Financial Economics*, 74, 209-235.
- Li, B. (2006a). A new approach to cluster analysis: the clustering-function-based method. *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, 68, 457-476.
- Li, B. (2006b). Sign eigenanalysis and its applications to optimization problems and robust statistics. *Computational Statistics & Data Analysis*, 50, 154-162.
- Lin, H.-Y. (2013). Feature selection based on cluster and variability analyses for ordinal multi-class classification problems. *Knowledge-Based Systems*, 37, 94-104.

- Lin, X., Yang, Z., & Song, Y. (2011). Intelligent stock trading system based on improved technical analysis and Echo State Network. *Expert Systems with Applications*, 38, 11347-11354.
- Linoff, G. S., & Berry, M. J. A. *Data mining techniques: for marketing, sales, and customer relationship management*: John Wiley & Sons.
- Liu, H., Motoda, H., Setiono, R., & Zhao, Z. Feature selection: An ever evolving frontier in data mining. In *Feature Selection in Data Mining* (pp. 4-13).
- Liu, H., & Setiono, R. (1996). A probabilistic approach to feature selection – a filter solution. In *In Proceedings of the 13th International Conference on Machine Learning* (pp. 319-327).
- Lu, T.-H. (2014). The profitability of candlestick charting in the Taiwan stock market. *Pacific-Basin Finance Journal*, 26, 65-78.
- Lu, T.-H., Shiu, Y.-M., & Liu, T.-C. (2012). Profitable candlestick trading strategies—The evidence from a new perspective. *Review of Financial Economics*, 21, 63-68.
- Majhi, R., Panda, G., & Sahoo, G. (2009). Development and performance evaluation of FLANN based model for forecasting of stock markets. *Expert Systems with Applications*, 36, 6800-6808.
- Maldonado, S., & Weber, R. (2009). A wrapper method for feature selection using Support Vector Machines. *Information Sciences*, 179, 2208-2217.
- Maldonado, S. n., & Weber, R. (2009). A wrapper method for feature selection using support vector machines. *Information Sciences*, 179, 2208-2217.
- Mehrotra, K., Mohan, C. K., & Ranka, S. (1997). *Elements of artificial neural networks*: MIT press.
- Menkhoff, L. (2010). The use of technical analysis by fund managers: International evidence. *Journal of Banking & Finance*, 34, 2573-2586.
- Min, J. H., & Lee, Y.-C. (2005). Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters. *Expert Systems with Applications*, 28, 603-614.
- Mitchell, T. M. (1997). *Machine Learning*: McGraw-Hill
- Mousavi, R., & Eftekhari, M. (2015). A new ensemble learning methodology based on hybridization of classifier ensemble selection approaches. *Applied Soft Computing*, 37, 652-666.
- Mozaffari, L., Mozaffari, A., & Azad, N. L. Vehicle speed prediction via a sliding-window time series analysis and an evolutionary least learning machine: A case study on San Francisco urban roads. *Engineering Science and Technology, an International Journal*.
- Mukherji, S., Dhatt, M. S., & Kim, Y. H. (1997). A Fundamental Analysis of Korean Stock Returns *Financial Analysts Journal*, 53, 75-80.
- Murthy, S. K. (1998). Automatic construction of decision trees from data: A multi-disciplinary survey. *Data mining and knowledge discovery*, 2, 345-389.
- Ng, W. W., Liang, X.-L., Li, J., Yeung, D. S., & Chan, P. P. (2014). LG-trader: Stock trading decision support based on feature selection by weighted Localized generalization error model. *Neurocomputing*.
- Omran, M., & Ragab, A. (2004). Linear Versus Non-linear Relationships Between Financial Ratios and Stock Returns: Empirical Evidence from Egyptian Firms. In *Review of Accounting and Finance* (Vol. 3, pp. 84 - 102).
- Ou, P., & Wang, H. (2009). Prediction of Stock Market Index Movement by Ten Data Mining Techniques. *Modern Applied Science*, 3, 28-42.
- Ou, P., & Wang, H. (2009). Prediction of stock market index movement by ten data mining techniques. *Modern Applied Science*, 3, 28-42.
- Oza, N. C., & Tumer, K. (2008). Classifier ensembles: Select real-world applications. *Information Fusion*, 9, 4-20.
- Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2014). Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Systems with Applications*.
- Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015a). Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques. *Expert Systems with Applications*, 42, 259-268.
- Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015b). Predicting stock market index using fusion of machine learning techniques. *Expert Systems with Applications*, 42, 2162-2172.
- Pereira, F., Mitchell, T., & Botvinick, M. (2009). Machine learning classifiers and fMRI: A tutorial overview. *NeuroImage*, 45, S199-S209.
- Perron, P. (1989). The great crash, the oil price shock, and the unit root hypothesis. *Econometrica: Journal of the Econometric Society*, 1361-1401.
- Piramuthu, S. (2004). Evaluating feature selection methods for learning in data mining applications. *European journal of operational research*, 156, 483-494.
- Priddy, K. L., & Keller, P. E. (2005). *Artificial neural networks: an introduction* (Vol. 68): SPIE press.

- Qian, B., & Rasheed, K. (2007). Stock market prediction with multiple classifiers. *Applied Intelligence*, 26, 25-33.
- Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, 1, 81-106.
- Ragg, T., Braun, H., & Landsberg, H. (1998). A comparative study of neural network optimization techniques. In *Artificial Neural Nets and Genetic Algorithms* (pp. 341-345): Springer.
- Rajabioun, R., Atashpaz-Gargari, E., & Lucas, C. (2008). Colonial competitive algorithm as a tool for Nash equilibrium point achievement. In *Computational science and its applications–iccsa 2008* (pp. 680-695): Springer.
- Rastrigin, L., & Erenstein, R. H. (1981). Method of Collective Recognition. *Energoizdat, Moscow*.
- Rizzo, G., d'Amato, C., Fanizzi, N., & Esposito, F. (2017). Tree-based models for inductive classification on the Web Of Data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 45, 1-22.
- Rokach, L. (2009). Taxonomy for characterizing ensemble methods in classification tasks: A review and annotated bibliography. *Computational Statistics & Data Analysis*, 53, 4046-4072.
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1988). Learning representations by back-propagating errors. *Cognitive modeling*, 5, 1.
- Sadka, G., & Sadka, R. (2009). Predictability and the earnings–returns relation. *Journal of Financial Economics*, 94, 87-106.
- Salzberg, S. (1994). C4.5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993. *Machine Learning*, 16, 235-240.
- Sarkheyli, A., Zain, A. M., & Sharif, S. (2015). Robust optimization of ANFIS based on a new modified GA. *Neurocomputing*.
- Schapire, R. E. (2001). The boosting approach to machine learning: an overview. in: *MSRI Workshop on Nonlinear Estimation and Classification, Berkeley, CA, USA*.
- Sepehri Rad, H., & Lucas, C. (2008). Application of imperialistic competition algorithm in recommender systems. In *13th Int'l CSI Computer Conference (CSICC'08), Kish Island, Iran*.
- Sharky, A. J. C., & Sharky, N. E. (1997). Combining diverse neural nets. *The Knowledge Engineering Review*, 12, 231-247.
- Shearer, C. (2000). The CRISP-DM model: the new blueprint for data mining. *Journal of data warehousing*, 5, 13-22.
- Shi, H. (2007). Best-first decision tree learning. *Annals of statistics*, 2, 337-407.
- Soliman, M.-T. (2008). The use of DuPont analysis by market participants. *The Accounting Review*, 83, 823-853.
- Svalina, I., Galzina, V., Lujčić, R., & Šimunović, G. (2013). An adaptive network-based fuzzy inference system (ANFIS) for the forecasting: The case of close price indices. *Expert Systems with Applications*, 40, 6055-6063.
- Tang, E. K., Suganthan, P. N., & Yao, X. (2006). An analysis of diversity measures. *Machine Learning*, 65, 247-271.
- Tsai, C.-F., & Hsiao, Y.-C. (2010). Combining multiple feature selection methods for stock prediction: Union, intersection, and multi-intersection approaches. *Decision Support Systems*, 50, 258-269.
- Tsai, C.-F., Hsu, Y.-F., & Yen, D. C. (2014). A comparative study of classifier ensembles for bankruptcy prediction. *Applied Soft Computing*, 24, 977-984.
- Tsai, C.-F., Lin, Y.-C., Yen, D. C., & Chen, Y.-M. (2011). Predicting stock returns by classifier ensembles. *Applied Soft Computing*, 11, 2452-2459.
- Tsai, C.-F., Lu, Y.-H., & Yen, D. C. (2012). Determinants of intangible assets value: The data mining approach. *Knowledge-Based Systems*, 31, 67-77.
- Tsymbol, A., Pechenizkiy, M., & Cunningham, P. (2005). Diversity in search strategies for ensemble feature selection. *Information Fusion*, 6, 83-98.
- Tumer, K., & Ghosh, J. (1996). Analysis of decision boundaries in linearly combined neural classifiers. *Pattern Recognition*, 29, 341-348.
- Vairappan, C., Tamura, H., Gao, S., & Tang, Z. (2009). Batch type local search-based adaptive neuro-fuzzy inference system (ANFIS) with self-feedbacks for time-series prediction. *Neurocomputing*, 72, 1870-1877.
- Wang, A., An, N., Chen, G., Li, L., & Alterovitz, G. (2015). Accelerating wrapper-based feature selection with K-nearest-neighbor. *Knowledge-Based Systems*.
- Wang, J.-L., & Chan, S.-H. (2006). Stock market trading rule discovery using two-layer bias decision tree. *Expert Systems with Applications*, 30, 605-611.
- Wang, L., & Wu, J. (2012). Neural Network Ensemble Model Using PPR and LS-SVR for Stock Market Forecasting. In D.-S. Huang, Y. Gan, V. Bevilacqua & J. Figueroa (Eds.), *Advanced Intelligent Computing* (Vol. 6838, pp. 1-8): Springer Berlin Heidelberg.

- Wen, Q., Yang, Z., Song, Y., & Jia, P. (2010). Automatic stock decision support system based on box theory and SVM algorithm. *Expert Systems with Applications*, 37, 1015-1022.
- Windeatt, T. (2005). Diversity measures for multiple classifier system analysis and design. *Information Fusion*, 6, 21-36.
- Witten, I. H., & Frank, E. (2011). *Data Mining: Practical Machine Learning Tools and Techniques* (3 ed.). San Francisco: Morgan Kaufmann.
- Woloszynski, T., & Kurzynski, M. (2011). A probabilistic model of classifier competence for dynamic ensemble selection. *Pattern Recognition*, 44, 2656-2668.
- Woloszynski, T., Kurzynski, M., Podsiadlo, P., & Stachowiak, G. W. (2012). A measure of competence based on random classification for dynamic ensemble selection. *Information Fusion*, 13, 207-213.
- Wolpert, D. (2001). The supervised learning no-free-lunch theorems. In: *Proceedings of the 6th Online World Conference on Soft Computing in Industrial Applications*, 25-42.
- Wozniak, M. (2008). Experiments on Linear Combiners. In E. Pietka & J. Kawa (Eds.), *Information Technologies in Biomedicine* (Vol. 47, pp. 445-452): Springer Berlin Heidelberg.
- Woźniak, M., Graña, M., & Corchado, E. (2014). A survey of multiple classifier systems as hybrid systems. *Information Fusion*, 16, 3-17.
- Wu, J.-L., Yu, L.-C., & Chang, P.-C. (2014). An intelligent stock trading system using comprehensive features. *Applied Soft Computing*, 23, 39-50.
- Ye, N. (2003). *The handbook of data mining*: CRC Press.
- Yu, H., Chen, R., & Zhang, G. (2014). A SVM Stock Selection Model within PCA. *Procedia Computer Science*, 31, 406-412.
- Yu, L., & Liu, H. (2003). Feature selection for high-dimensional data: A fast correlation-based filter solution. In *Proceedings of the 20th international conference on machine learning (ICML-03)* (pp. 856-863).
- Yunos, Z. M., Shamsuddin, S. M., & Sallehuddin, R. (2008). Data Modeling for Kuala Lumpur Composite Index with ANFIS. In *Modeling & Simulation, 2008. AICMS 08. Second Asia International Conference on* (pp. 609-614): IEEE.
- Zhang, G., Patuwo, B. E., & Hu, M. Y. (1998). Forecasting with artificial neural networks:: The state of the art. *International journal of forecasting*, 14, 35-62.
- Zhang, X., Hu, Y., Xie, K., Wang, S., Ngai, E., & Liu, M. (2014). A causal feature selection algorithm for stock prediction modeling. *Neurocomputing*.
- Zhang, X., Hu, Y., Xie, K., Zhang, W., Su, L., & Liu, M. (2014). An Evolutionary Trend Reversion Model for Stock Trading Rule Discovery. *Knowledge-Based Systems*.
- Zhu, Y., & Shasha, D. (2003). Efficient elastic burst detection in data streams. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 336-345): ACM.
- Zhu, Y., & Zhou, G. (2009). Technical analysis: An asset allocation perspective on the use of moving averages. *Journal of Financial Economics*, 92, 519-544.

Appendix A.

Probabilistic Back-propagation

One of the common training methods is back propagation (Rumelhart, Hinton, & Williams, 1988). In this approach, first, the input features are fed into the network and propagated through it to compute function output. At the end of the first step, associated loss function with the network structure parameters is determined. Then, in the second phase, derivatives of training loss are propagated backward from the output layer to the input. The derivative function is used to update the connection weights between network nodes, e.g., stochastic gradient decent. In probabilistic back propagation approach, instead of using point estimates for the synaptic weights in the network, a collection of one-dimensional Gaussians that each one approximating the marginal posterior distribution of different weight, is implemented.

PBP also has two phases equivalent to the ones of BP. In the first phase, the input data is propagated forward through the network. However, since the weights are now random, the activations produced in each layer are also random and result in (intractable) distributions. PBP sequentially approximates each of these distributions with a collection of one-dimensional Gaussians that match their marginal means and variances. At the end of this phase, PBP computes, instead of the prediction error, the logarithm of the marginal probability of the target variable. In the second phase, the gradients of this quantity concerning the means and variances of the approximate Gaussian posterior are propagated back using reverse-mode differentiation as in classic back propagation. These derivatives are finally used to update the means and variances of the posterior approximation. The update rule used by PBP is not the standard step in the direction of the gradient of the loss made by the classic backpropagation algorithm. More detailed of PBP approach is provided in (Hernández-Lobato & Adams, 2015).

ID3 (Iterative Dichotomiser 3)

In decision tree models, ID3 which is invented by Ross Quinlan (Quinlan, 1986) used to generate a decision tree from a dataset. The ID3 algorithm starts with root node which contains with the original set S . On each iteration of the algorithm; it searches among attributes to calculate the entropy (or information ratio) of that attribute for the set S . It then selects the attribute which has the smallest entropy (or largest information ratio) value. After that, the set S is split by the best attribute (which reduces maximum entropy), to produce subsets of the data. The splitting subsets continues to recur on each subset, considering only attributes never selected before.

C4.5 Decision Tree

In C4.5 decision tree method, same concept of reduction entropy is used for splitting the dataset. At each node of the tree, C4.5 chooses the attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other. The splitting criterion is the normalized information gain. The attribute with the highest normalized information gain is chosen to make the decision. C4.5 in addition to ID3, accepts both continuous and discrete features; handles incomplete data points; solves over-fitting problem by bottom-up technique (pruning); and different weights can be applied the features.

Rough Decision Trees (RDT) Model

Rough set theory applies the unclear relation and data pattern comparison based on the concept of an information system with indiscernible data, where the data is uncertain or inconsistent. The data is grouped into classes called elementary sets. A typical structure of RDT which is shown Figure A.1, includes the following steps:

1. Input the training data set T1.
2. Discretize the numeric or continuous attributes if any, and label the modified data set as T2.
3. Obtain the minimal decision relative reduct of T2, say R.
4. Reduce T2 based on reduct R and label reduced data set as T3.
5. Apply ID3 algorithm on T3 to induce decision tree.
6. If needed, convert decision tree to rules by traversing all the possible paths from root to each leaf node.

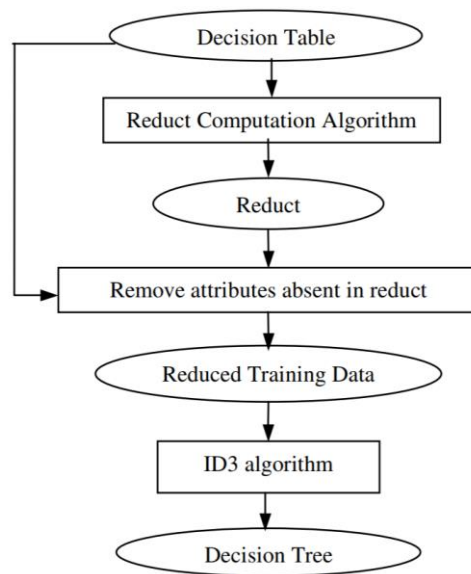


Figure 8A.1 Rough Decision Trees (RDT) structure

Appendix B.

CAPM Model

- β coefficient is the amount of changes in the stock return to market and accounted as follow.

$$\beta = (\text{Cov}(\text{Market return} * \text{stock return})) / \text{Var}(\text{Market return}). \quad (A_1)$$

- Market expected return, shows the amount of market return in a definite time which is gained with this formula:

$$r_m = \frac{P_t - P_{t-1}}{P_{t-1}} \quad (A_2)$$

In this p_t = the market indicator at the end of period (for example 2013/12/28) and P_{t-1} = the market indicator at the beginning of period (for example 2013/1/1).

- Return without risk also can be done through this formula.

$$r_{ft} = \frac{(p_t - p_{t-1}) + D_t}{p_{t-1}} \times 100 \quad (A_3)$$

In this p_t is end of period stock price, P_{t-1} = beginning of period stock price and D_t = Benefits of stock ownership which has belonged to Shareholder in period t. if we have capital increase in period of investment from savings or receivables and cash income then the formula will change as a follow :

$$r_{ft} = \frac{D_t + p_t(1 + \alpha + \beta) - (p_{t-1} + c\alpha)}{p_{t-1} + c\alpha} \times 100 \quad (A_4)$$

In this α = the percent of capital increase of the receivables and cash income, β = the percent of capital increase of savings, c = Nominal amount paid by investor to increase the capital of the receivables and cash income. We use this formula for calculate the r_m , r_f and β .

Appendix C.

Feature concepts (formula)

Current ratio is a liquidity ratio that measures a company's ability to pay short-term obligations ($\text{Current Ratio} = \text{Current Assets} / \text{Current Liabilities}$).

Quick ratio is an indicator of a company's short-term liquidity position and measures a company's ability to meet its short-term obligations with its most liquid assets ($\text{Quick ratio} = (\text{Current assets} - \text{inventory} - \text{prepaid expenses}) / \text{Current Liabilities}$).

Net working capital is a measure of a company's liquidity, operational efficiency and its short-term financial health ($\text{Current Ratio} = \text{Current Assets} - \text{Current liabilities}$).

Average payment period is a solvency ratio that measures the average number of days it takes a business to pay its vendors for purchases made on credit ($\text{Average Payment Period} = \text{Average Accounts Payable} / (\text{Total Credit Purchases} / \text{Days})$).

Assets turnover ratio measures the value of a company's sales or revenues relative to the value of its assets ($\text{Asset Turnover} = \text{Total Sales} / (\text{Beginning Assets} + \text{Ending Assets}) / 2$).

Equity ratio shows how much of the company's assets are funded by equity shares ($\text{Equity Ratio} = \text{Total shareholder equity} / \text{Total assets}$).

Debt coverage ratio measures how well a company can pay its entire debt service ($\text{Debt coverage ratio} = \text{Net Operating Income} / \text{Total Debt}$).

Return on asset (after tax) is an indicator of how profitable a company is relative to its total assets ($\text{Return on assets (ROA)} = \text{Net Income} / \text{Total assets}$).

Beta (β) is a measure of the volatility, or systematic risk, of an individual stock in comparison to the unsystematic risk of the entire market.

EPS, earning per share of stock.

Dividend per share (DPS) is the sum of declared dividends issued by a company for every ordinary share outstanding.

P/E = Price / EPS

P/S = Price / Sales

Company book value represents the total amount a company is worth if all its assets are sold and all the liabilities are paid back which is not equal to the **Market value** that is evaluated by the values of companies stock in the market.

List of Tables

Table 3-1 Stock pricing models	25
Table 3-2 Financial input features	26
Table 3-3 Confusion matrix.....	29
Table 3-4 Confusion matrix for five classes.....	29
Table 3-5 Algorithms Comparison for real return variable	33
Table 3-6 Algorithm comparison for risk variable	36
Table 3-7 Weighting for risk parameter	37
Table 3-8 Weighting for real return parameter.....	38
Table 3-9 Selected features for risk and real return parameters	40
Table 3-10 Algorithms Deviation.....	40
Table 3-11 Comparison results with other studies.....	42
Table 4-1 Meta cost penalty score matrix.....	54
Table 4-2 Confusion matrix for a five-class prediction problem.....	54
Table 4-3 SSCA calculations on different number of clusters	60
Table 4-4 Average accuracy percentage of selected sets for stock return prediction (without diversity).	61
Table 4-5 Average accuracy percentage of selected sets for stock risk prediction (without diversity).	61
Table 4-6 Average accuracy percentage of stock return prediction with diversity creation.....	62
Table 4-7 Prediction results of the best ensemble for real return prediction.	62
Table 4-8 Average accuracy percentage of stock risk prediction with diversity creation.	63
Table 4-9 Prediction results of the best ensemble for risk prediction.....	63
Table 4-10 Selected features for real return prediction with Wrapper-GA algorithm.....	65
Table 4-11 Selected features for risk prediction with Wrapper-GA algorithm.	67
Table 4-12 Comparison of the proposed fusion scheme versus other studies	68
Table 5-1 Raw approach.....	84
Table 5-2 Signal approach.....	84
Table 5-3 Details of the applied data sets	85
Table 5-4 The correct prediction percentage for 1 and 6 day periods with total number of buy and sell signals	87
Table 5-5 Optimal features and their costs in second approach	91
Table 5-6 Comparison between algorithms	92

Summary list of my publications during the PhD

Fusion of multiple diverse predictors in stock market

S Barak, S Ortobelli

Information Fusion 36, 90-102

Wrapper ANFIS-ICA method to do stock market timing and feature selection on the basis of Japanese Candlestick

S Barak, J Heidary dahooie, T Tichý

Expert Systems with Applications 42 (23), 9221-9235

Developing an approach to evaluate stocks by forecasting effective features with data mining methods

S Barak, M Modarres

Expert Systems with Applications 42 (3), 1325-1339

Dependency evaluation of financial market returns for classifying and grouping stocks

S Barak

Intelligent Systems and Signal Processing (ICSPIS), 2017 3rd Iranian ...

A novel hybrid fuzzy DEA-Fuzzy MADM method for airlines safety evaluation

S Barak, J Heidary

Journal of Air Transport Management 73, 134-149

Declaration of Authorship

I, Sasan Barak, declare that:

- I was (a) aware of the fact that my dissertation is fully covered by Act No. 121/2000 Coll.- Copyright Act, Particularly Sec. 35 - use of the work within civil and religious ceremonies, in school performances and school use of the work and Sec. 60 - school work;
- acknowledge that the VSB -Technical University of Ostrava (the VSB - TUO) has the right to use the dissertation in a non-profit manner for its internal needs (Sec. 35 para.3);
- I agree that the dissertation will be electronically archived at the Central Library of VSB -TUO and one copy shall be filed with the dead of the dissertation work. I agree that bibliographic data on the dissertation will be published in the information system of VSB - TUO;
- It was agreed that I shall conclude a license agreement with VSB - TUO, in case of interest on its part, containing permission to use the work in accordance with Sec.12 para.4 of the Copyright Act;
- It was agreed I can use the work, dissertation work, or license its use only with the approval of VSB - TUO, which is authorized in this case to ask me to make a reasonable contribution to the costs that were incurred by VSB - TUO towards the creation of the work (up to the actual amount).

Signed: *Sasan Barak*

Place and date: May 2019